

Lección 3a

procesos, periférico, *drivers* y servicios ampliados

Diseño de Sistemas Operativos
Grado en Ingeniería Informática

Lecturas recomendadas

Base



1. Carretero 2007:
 1. Cap.7

Recomendada

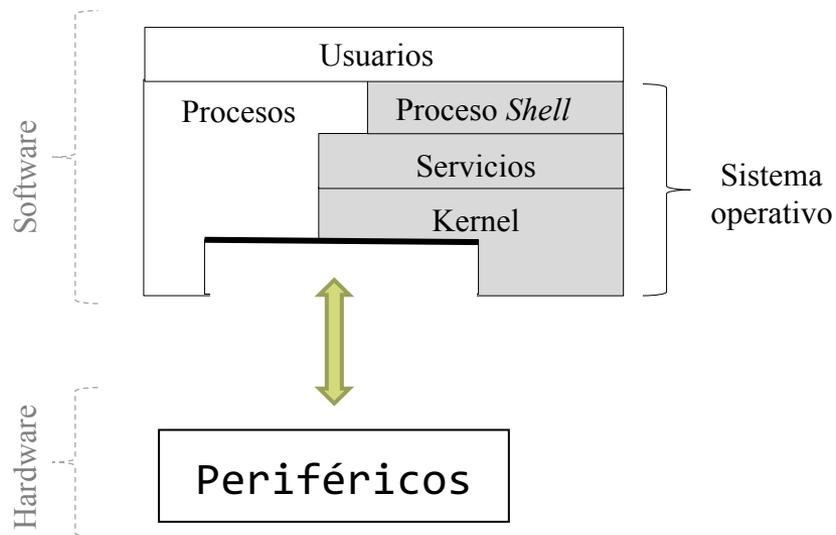


1. Tanenbaum 2006(en):
 1. Cap.3
2. Stallings 2005(en):
 1. Parte tres
3. Silberschatz 2006:
 1. Cap. Sistemas de E/S

A recordar...

1. Estudiar la teoría asociada.
 - ▶ Estudiar el material asociado a la bibliografía: las transparencias solo no son suficiente.
2. Repasar lo visto en clase.
 - ▶ Realizar el cuaderno de prácticas progresivamente.
3. Ejercitar las competencias.
 - ▶ Realizar las prácticas progresivamente.
 - ▶ Realizar todos los ejercicios posibles.

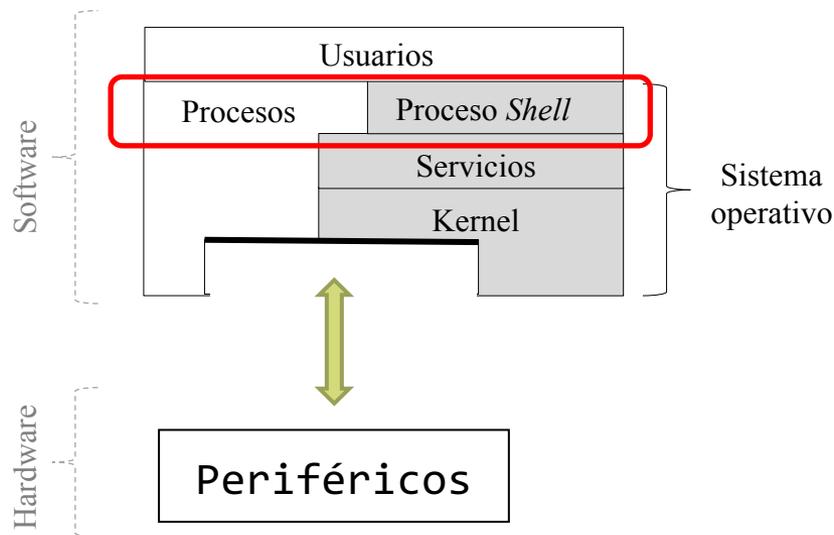
Contenidos



► **Procesos**

► **Periféricos**

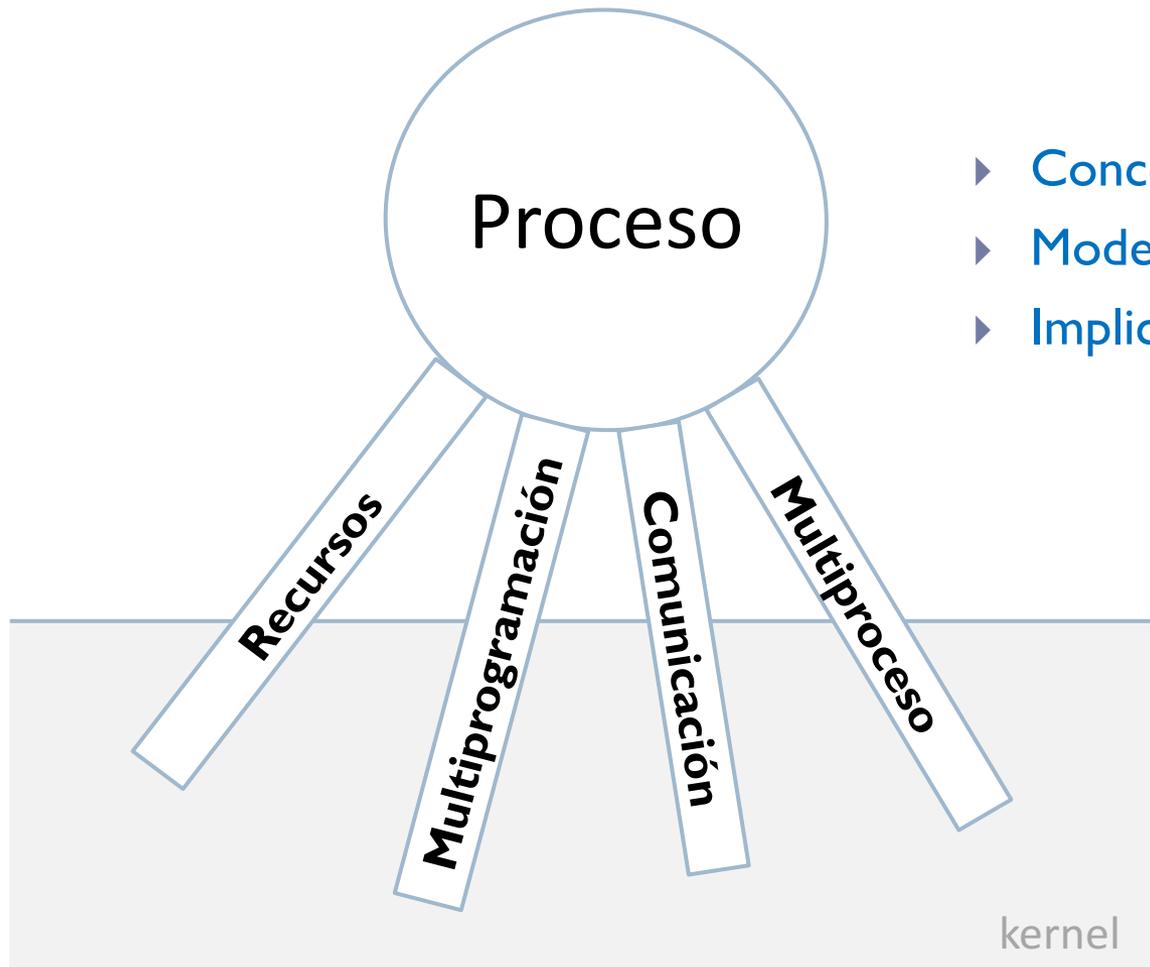
Contenidos



► **Procesos**

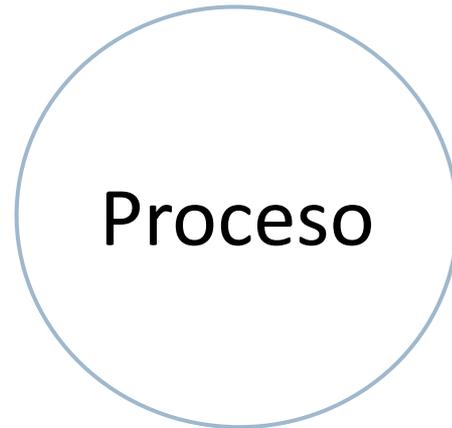
► **Periféricos**

Introducción



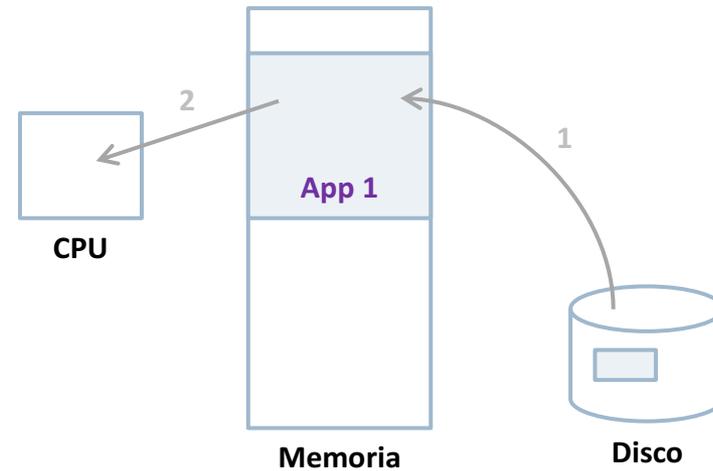
- ▶ Concepto de proceso
- ▶ Modelo ofrecido
- ▶ Implicaciones en S.O.

Introducción



- ▶ **Concepto de proceso**

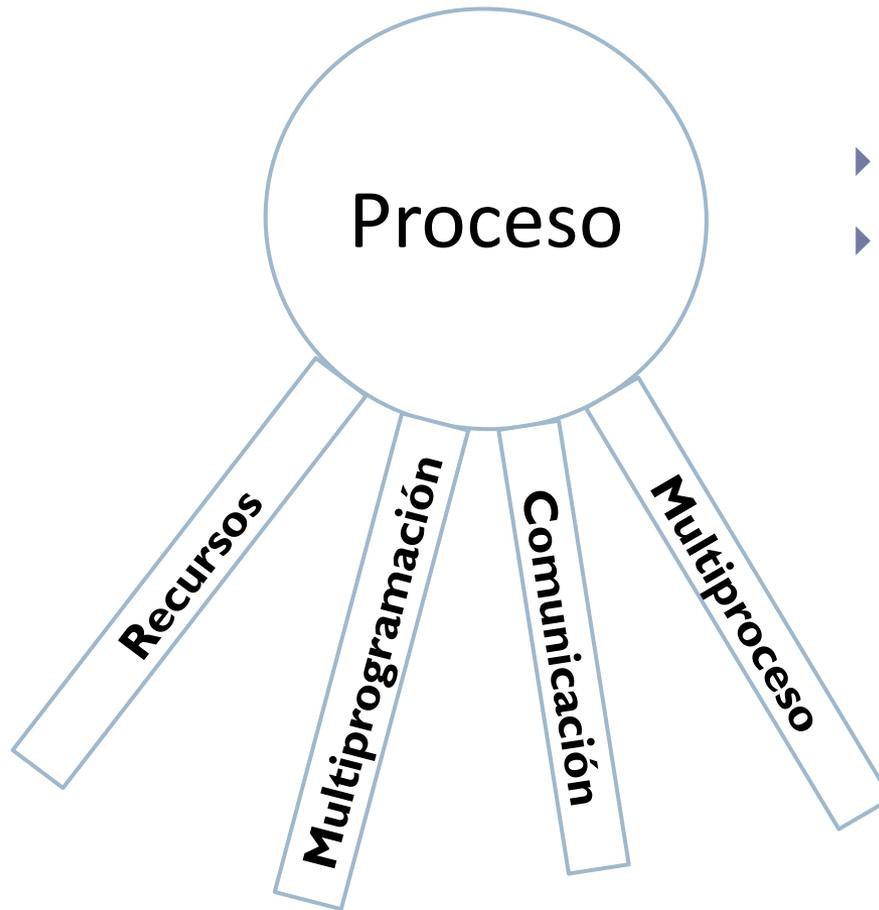
Concepto de proceso



▶ Proceso

- ▶ Programa en ejecución
- ▶ Unidad de procesamiento gestionada por el S.O.

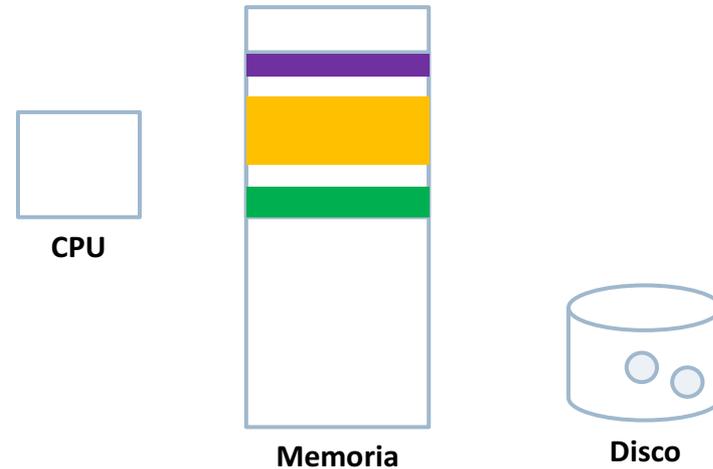
Introducción



- ▶ Concepto de proceso
- ▶ **Modelo ofrecido**

Modelo ofrecido

- recursos
- multiprogramación
 - protección/compartición
 - jerarquía de procesos
- multitarea
- multiproceso

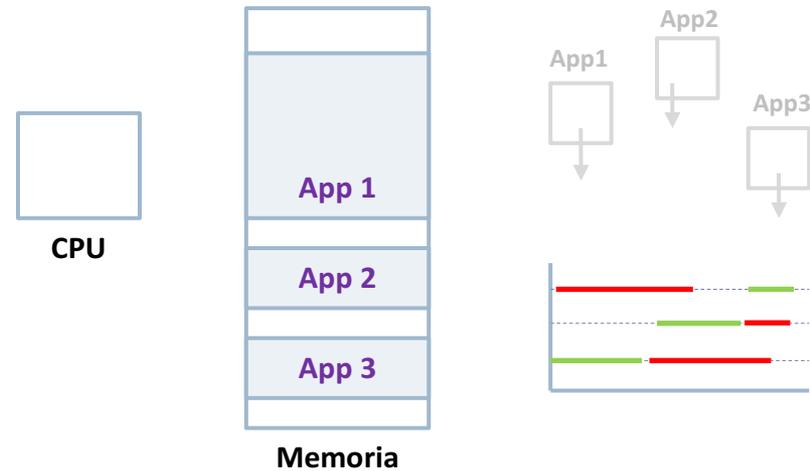


▶ Recursos asociados

- ▶ Zonas de memoria
 - ▶ Al menos: código, datos y pila
- ▶ Archivos abiertos
- ▶ Señales

Modelo ofrecido

- recursos
- **multiprogramación**
 - protección/compartición
 - jerarquía de procesos
- multitarea
- multiproceso

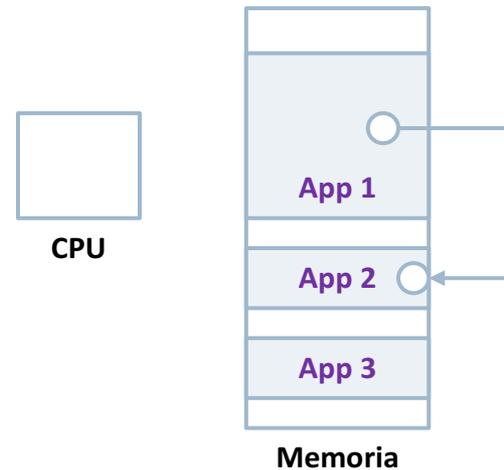


▶ Multiprogramación

- ▶ Tener varias aplicaciones en memoria
- ▶ Si una aplicación se bloquea por E/S, entonces se ejecuta mientras otra hasta que quede bloqueada
 - ▶ Cambio de contexto voluntario (C.C.V.)
- ▶ Eficiencia en el uso del procesador
- ▶ Grado de multiprogramación = número de aplicaciones en RAM

Modelo ofrecido

- recursos
- multiprogramación
 - **protección/compartición**
 - jerarquía de procesos
- multitarea
- multiproceso

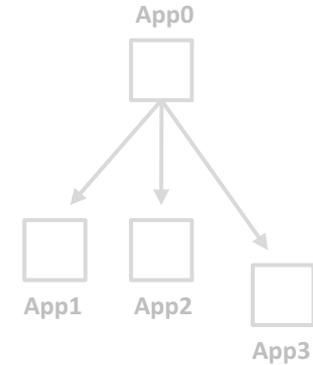
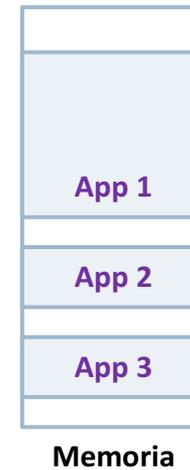


► Protección / Compartición

- El espacio de direcciones privado por aplicación, pero
- Posibilidad de comunicar datos entre dos aplicaciones
 - Paso de mensajes
 - Compartición de memoria

Modelo ofrecido

- recursos
- multiprogramación
 - protección/compartición
 - **jerarquía de procesos**
- multitarea
- multiproceso

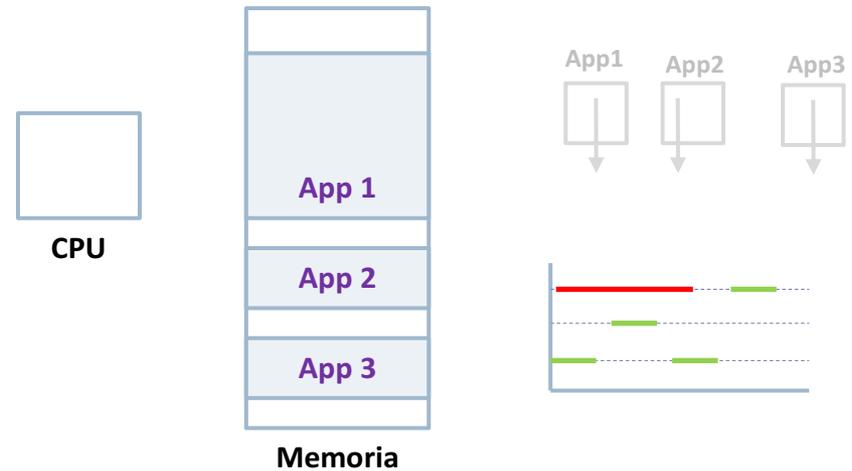


▶ Jerarquía de procesos

- ▶ Creación de proceso
 - ▶ Como copia de otro proceso existente
 - ▶ A partir del programa en disco
 - ▶ Como proceso en el arranque
- ▶ Grupo de procesos que comparten mismo tratamiento

Modelo ofrecido

- recursos
- multiprogramación
 - protección/compartición
 - jerarquía de procesos
- **multitarea**
- multiproceso

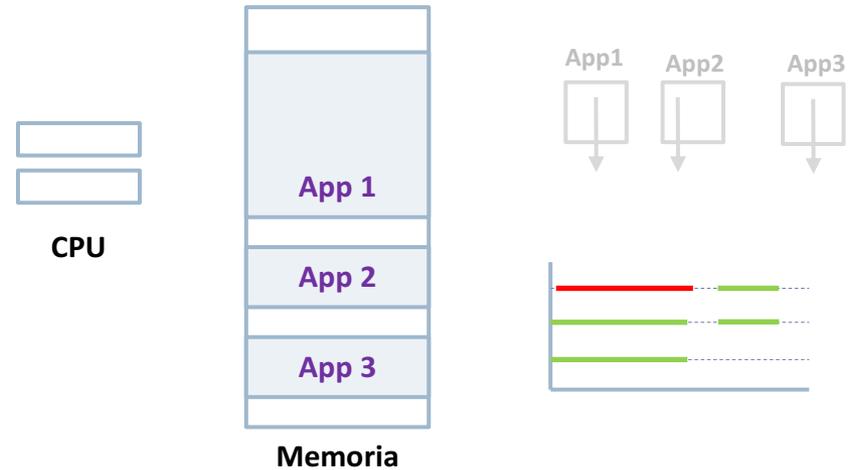


▶ **Multitarea**

- ▶ Cada proceso se ejecuta un quantum de tiempo (Ej.: 5 ms) y se rota el turno para ejecutar procesos no bloqueados
 - ▶ Cambio de contexto involuntario (C.C.I.)
- ▶ Reparto del uso del procesador
 - ▶ Parece que todo se ejecuta a la vez

Modelo ofrecido

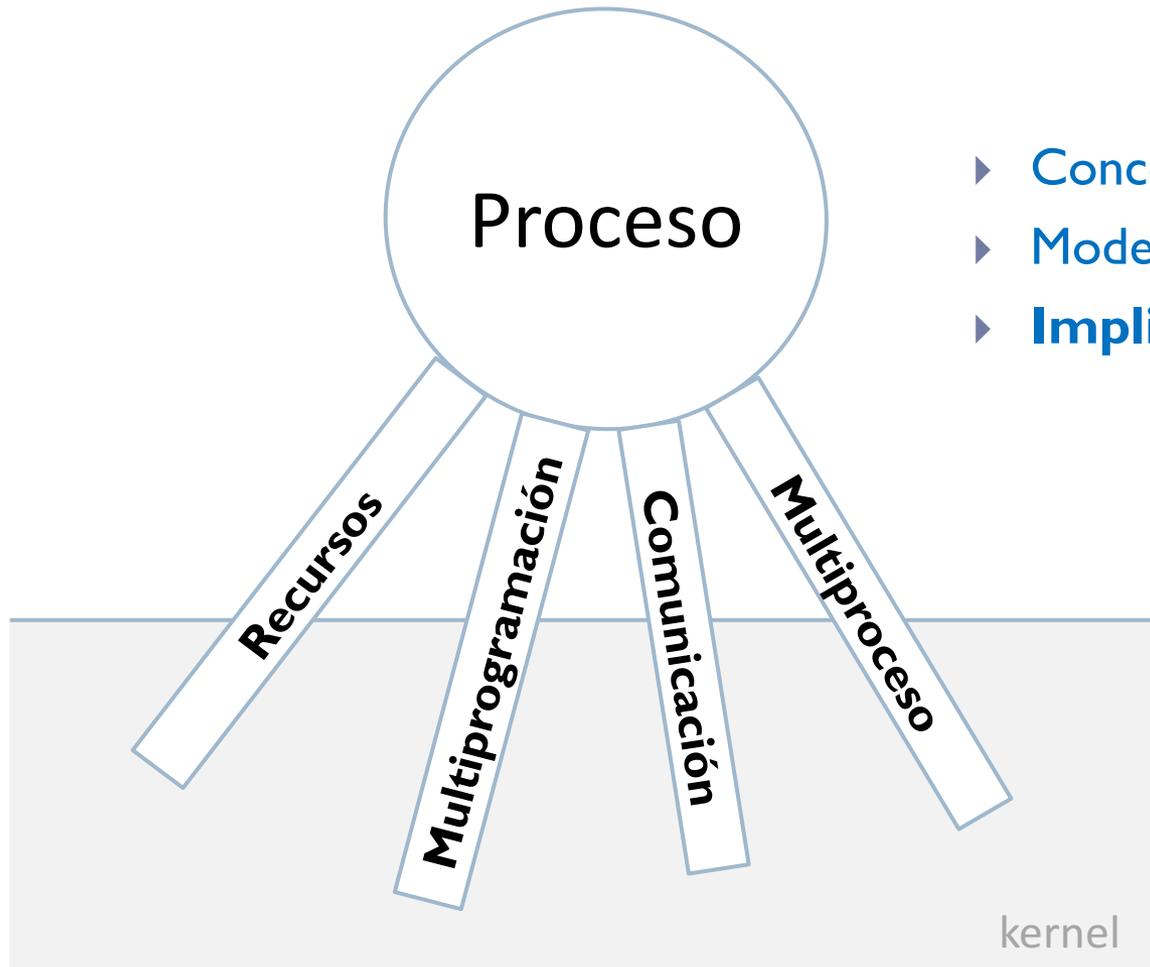
- recursos
- multiprogramación
 - protección/compartición
 - jerarquía de procesos
- multitarea
- **multiproceso**



► Multiproceso

- Se dispone de varios procesadores (multicore/multiprocesador)
- Además del reparto de cada CPU (multitarea) hay paralelismo real entre varias tareas (tantas como procesadores)
 - Se suele usar planificador y estructuras de datos separadas por procesador con algún mecanismo de equilibrio de carga

Introducción



- ▶ Concepto de proceso
- ▶ Modelo ofrecido
- ▶ **Implicaciones en S.O.**

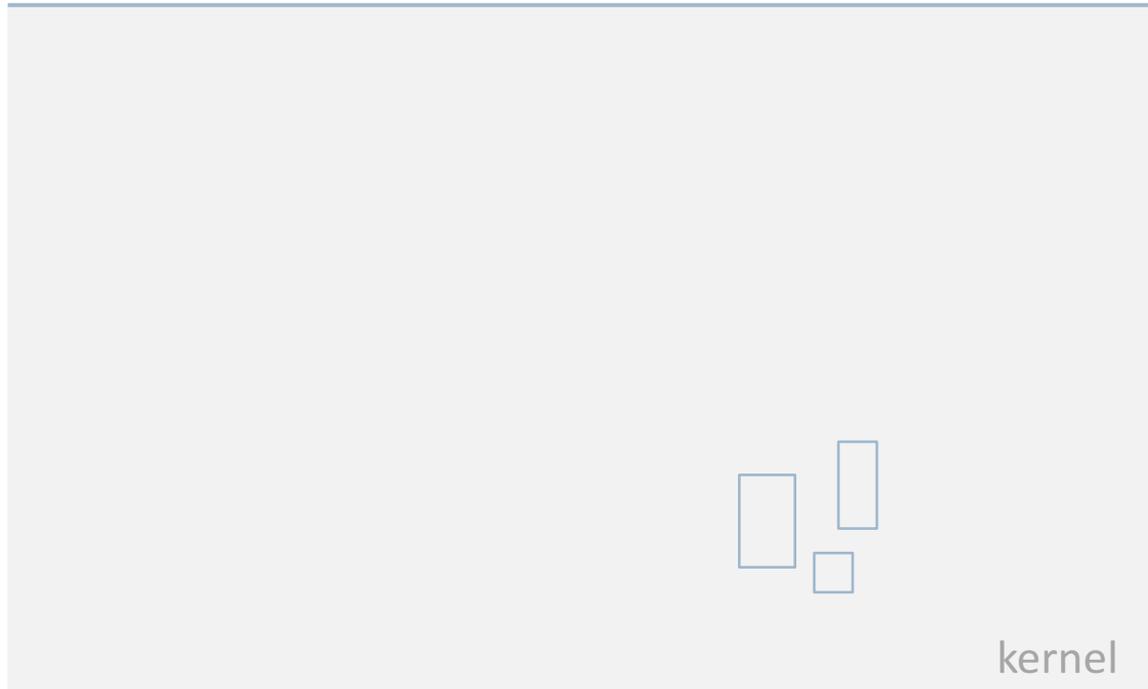
Implicaciones en el sistema operativo

I. Estructuras de datos

| Requisitos | Información (en estructuras de datos) |
|--------------------------------|---|
| Recursos | <ul style="list-style-type: none">• Zonas de memoria (código, datos y pila)• Archivos abiertos• Señales activas |
| Multiprogramación | <ul style="list-style-type: none">• Estado de ejecución• Contexto: registros de CPU...• Lista de procesos |
| ○ Protección / Compartición | <ul style="list-style-type: none">• Paso de mensajes<ul style="list-style-type: none">• Cola de mensajes de recepción• Memoria compartida<ul style="list-style-type: none">• Zonas, locks y conditions |
| ○ Jerarquía de procesos | <ul style="list-style-type: none">• Relación de parentesco• Conjuntos de procesos relacionados• Procesos de una misma sesión |
| Multitarea | <ul style="list-style-type: none">• Quantum restante• Prioridad |
| Multiproceso | <ul style="list-style-type: none">• Afinidad |

Implicaciones en el sistema operativo

I. Estructuras de datos



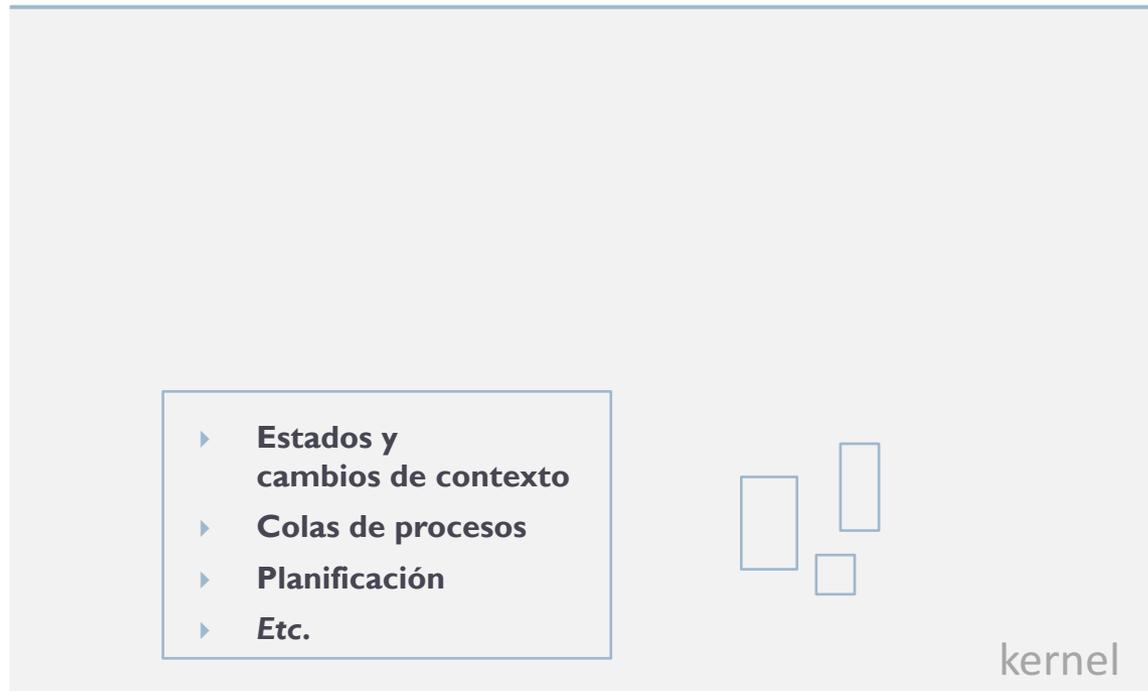
Implicaciones en el sistema operativo

2. Funciones: de gestión internas

| Requisitos | Información (en estructuras de datos) | Funciones (internas, servicio y API) |
|--------------------------------|---|--|
| Recursos | <ul style="list-style-type: none">• Zonas de memoria (código, datos y pila)• Archivos abiertos• Señales activas | <ul style="list-style-type: none">• Diversas funciones internas• Diversas funciones de servicio para memoria, ficheros, etc. |
| Multiprogramación | <ul style="list-style-type: none">• Estado de ejecución• Contexto: registros de CPU...• Lista de procesos | <ul style="list-style-type: none">• Int. hw/sw de dispositivos• Planificador• Crear/Destruir/Planificar proceso |
| ○ Protección / Compartición | <ul style="list-style-type: none">• Paso de mensajes<ul style="list-style-type: none">• Cola de mensajes de recepción• Memoria compartida<ul style="list-style-type: none">• Zonas, locks y conditions | <ul style="list-style-type: none">• Envío/Recepción mensaje y gestión de la cola de mensaje• API concurrencia y gestión de estructuras de datos |
| ○ Jerarquía de procesos | <ul style="list-style-type: none">• Relación de parentesco• Conjuntos de procesos relacionados• Procesos de una misma sesión | <ul style="list-style-type: none">• Clonar/Cambiar imagen de proceso• Asociar procesos e indicar proceso representante |
| Multitarea | <ul style="list-style-type: none">• Quantum restante• Prioridad | <ul style="list-style-type: none">• Int. hw/sw de reloj• Planificador• Crear/Destruir/Planificar proceso |
| Multiproceso | <ul style="list-style-type: none">• Afinidad | <ul style="list-style-type: none">• Int. hw/sw de reloj• Planificador• Crear/Destruir/Planificar proceso |

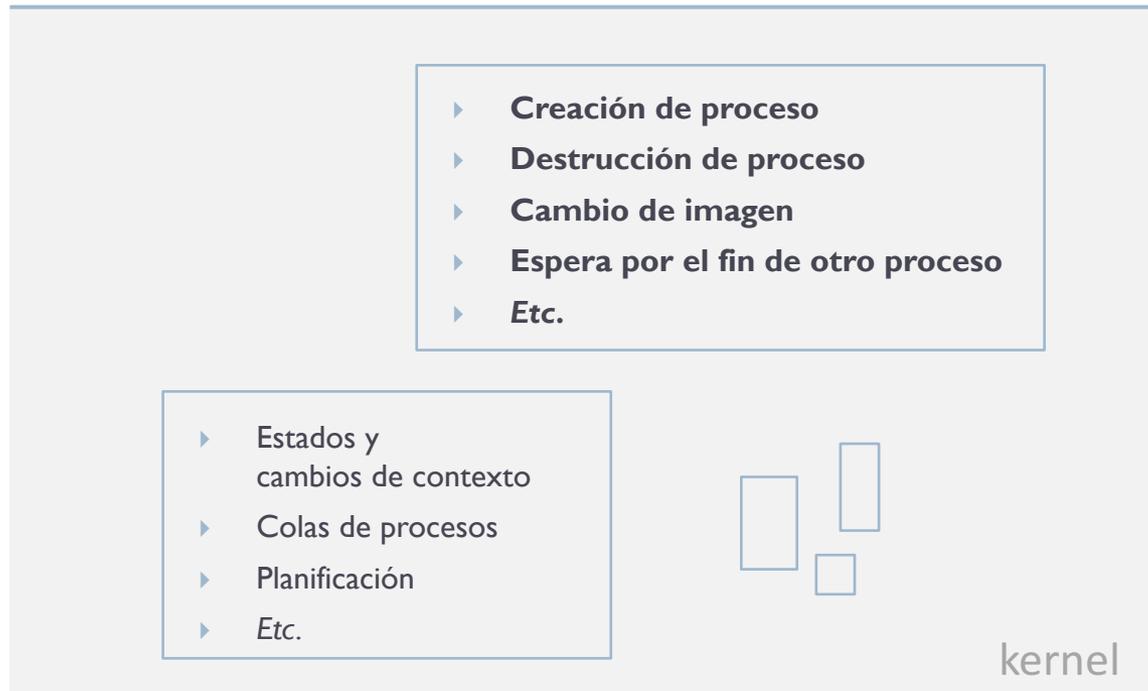
Implicaciones en el sistema operativo

2. Funciones: de gestión internas



Implicaciones en el sistema operativo

3. Funciones: de servicio



Implicaciones en el sistema operativo

3. Funciones: API de servicio

- ▶ `fork, exit, exec, wait, ...`
- ▶ `pthread_create, pthread...`

- ▶ Creación de proceso
- ▶ Destrucción de proceso
- ▶ Cambio de imagen
- ▶ Espera por el fin de otro proceso
- ▶ *Etc.*

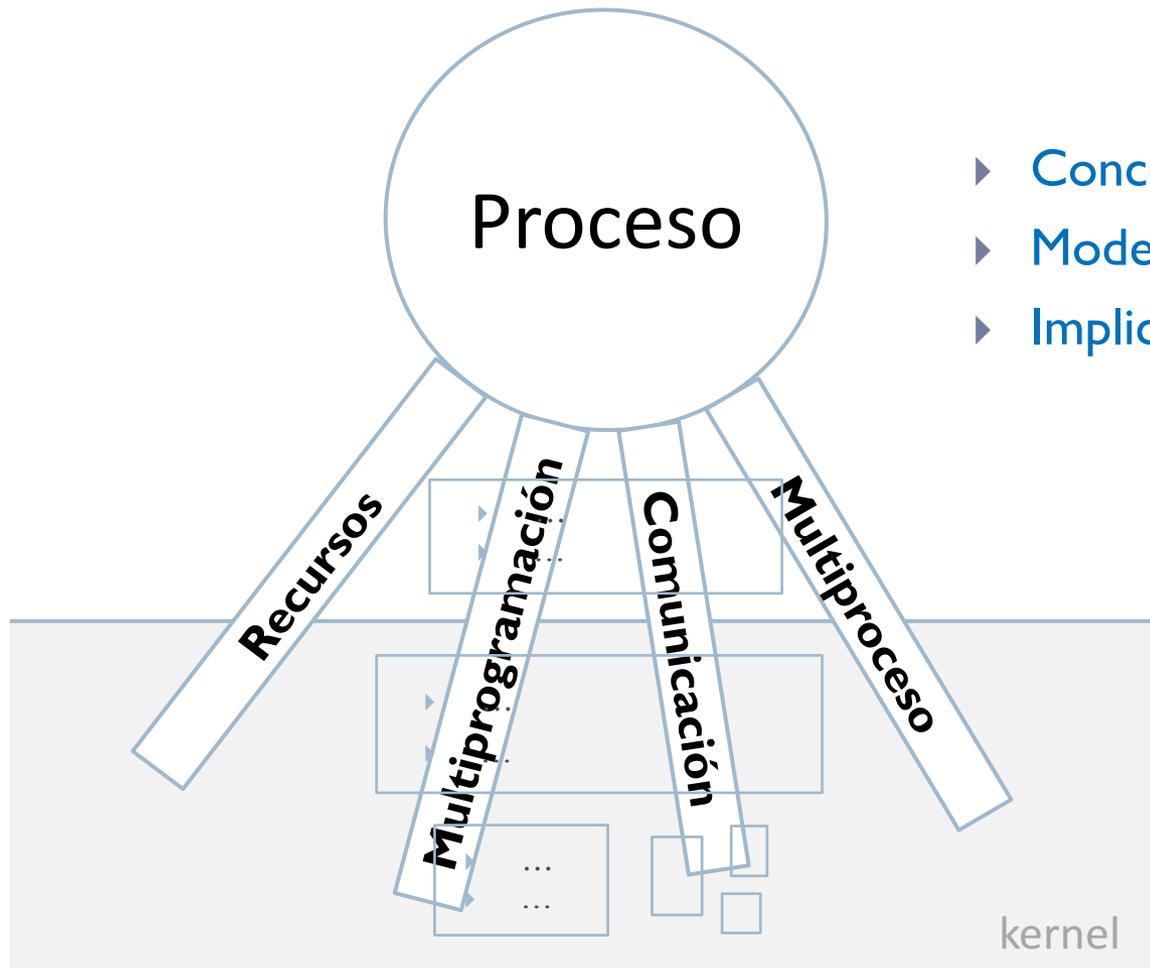
- ▶ Estados y cambios de contexto
- ▶ Colas de procesos
- ▶ Planificación
- ▶ *Etc.*



kernel

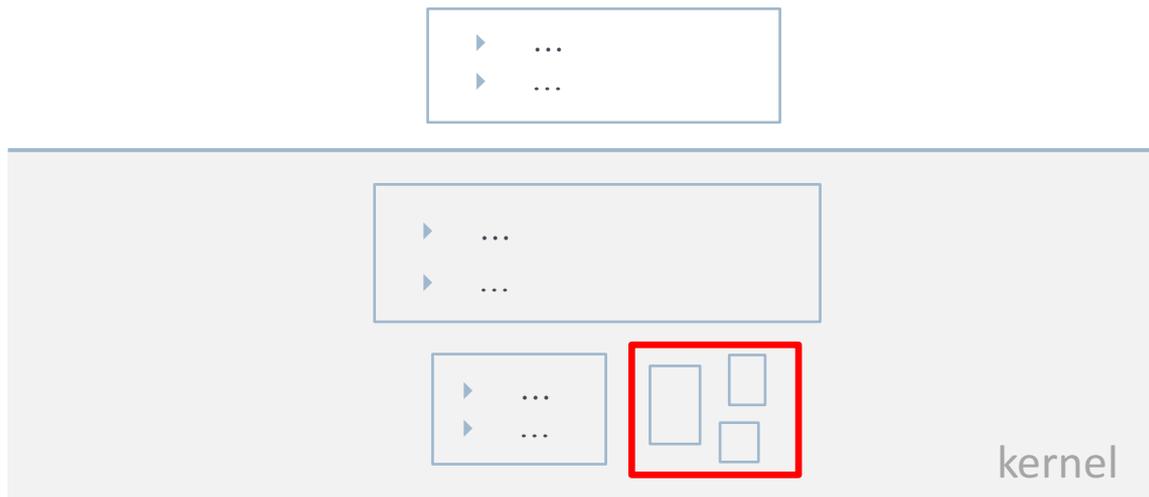
Introducción

resumen

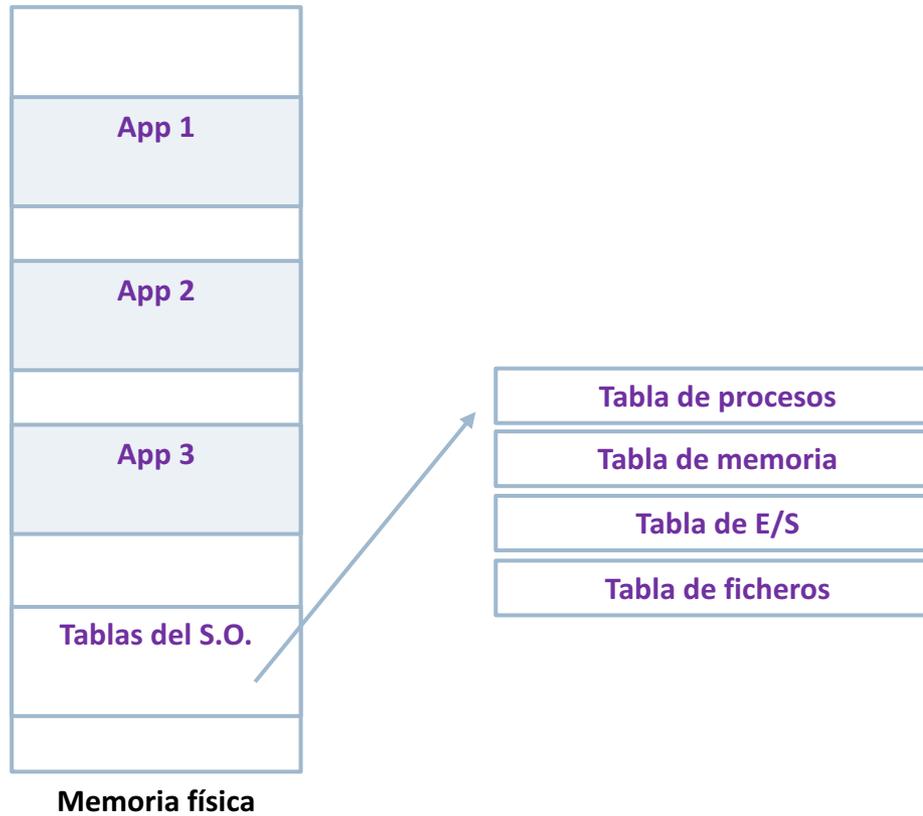


- ▶ Concepto de proceso
- ▶ Modelo ofrecido
- ▶ Implicaciones en S.O.

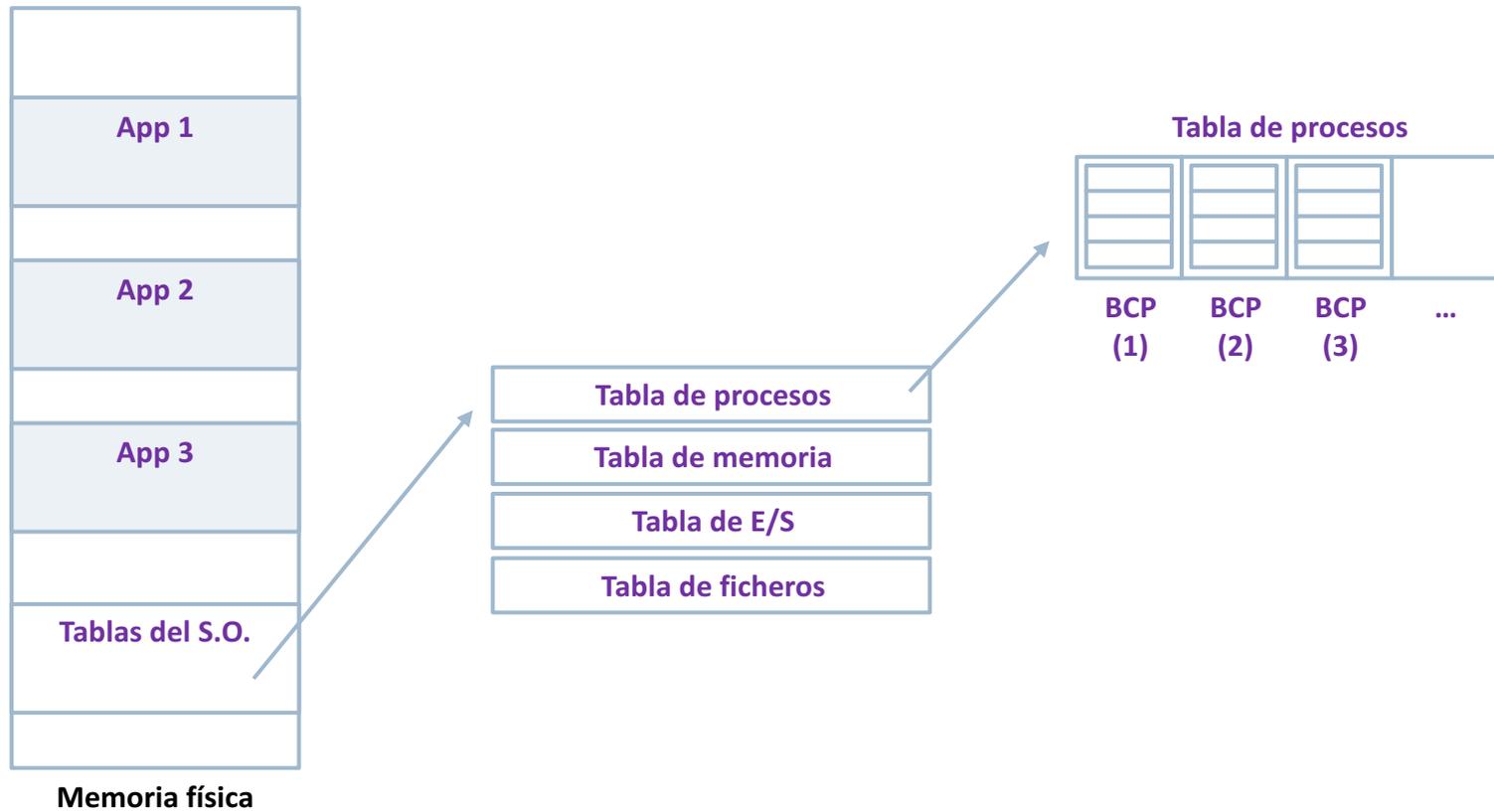
Principales estructuras de datos



Información en el sistema operativo



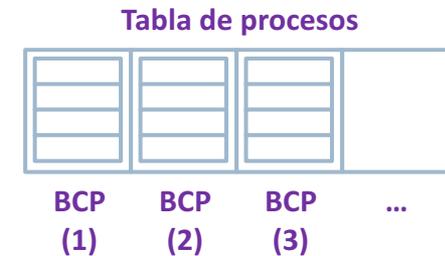
Información para un proceso



BCP: entrada de la tabla de procesos

▶ Gestión de proceso

| | |
|---------|---------------------------------|
| estado | ▶ Registros generales |
| | ▶ Contador de programa |
| | ▶ Registro de estado |
| | ▶ Puntero de pila |
| Id. | ▶ Identificador del proceso |
| | ▶ Proceso padre |
| | ▶ Grupo de proceso |
| gestión | ▶ Prioridad |
| | ▶ Parámetros del planificador |
| | ▶ Señales |
| | ▶ Instante inicio de ejecución |
| | ▶ Tiempo de uso de CPU |
| | ▶ Tiempo hasta siguiente alarma |



▶ *Process Control Block* (PCB / BCP)

- ▶ Estructura de datos con la información necesaria para gestionar un proceso en particular
- ▶ Manifestación de un proceso en el kernel

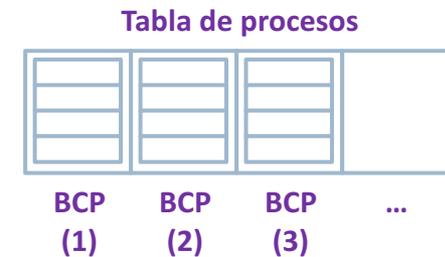
▶ *Thread Control Block* (TCB / BCT)

- ▶ Similar al BCP para cada hilo de un proceso

BCP: entrada de la tabla de procesos

▶ Gestión de proceso

| | |
|---------|---------------------------------|
| estado | ▶ Registros generales |
| | ▶ Contador de programa |
| | ▶ Registro de estado |
| | ▶ Puntero de pila |
| Id. | ▶ Identificador del proceso ← |
| | ▶ Proceso padre |
| | ▶ Grupo de proceso |
| gestión | ▶ Prioridad |
| | ▶ Parámetros del planificador |
| | ▶ Señales |
| | ▶ Instante inicio de ejecución |
| | ▶ Tiempo de uso de CPU |
| | ▶ Tiempo hasta siguiente alarma |



▶ *Process Identification* (PID)

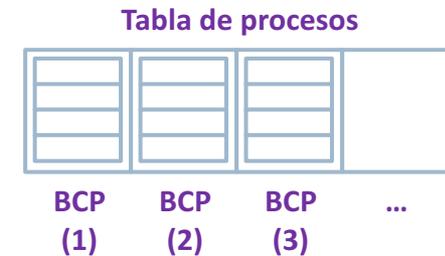
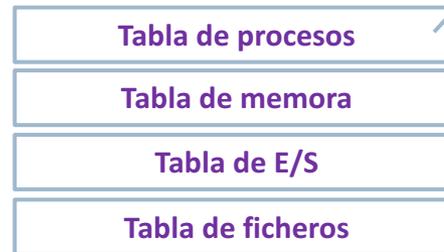
- ▶ Identificador de cara a los usuarios
- ▶ Número positivo de 16 bits (32767) dinámicamente asignado, reusado no de forma inmediata

▶ *Address of process descriptor* (APD)

- ▶ Identificación dentro del kernel
- ▶ Existe mecanismo PID -> APD (Ej.: hash)

Dónde: información del proceso

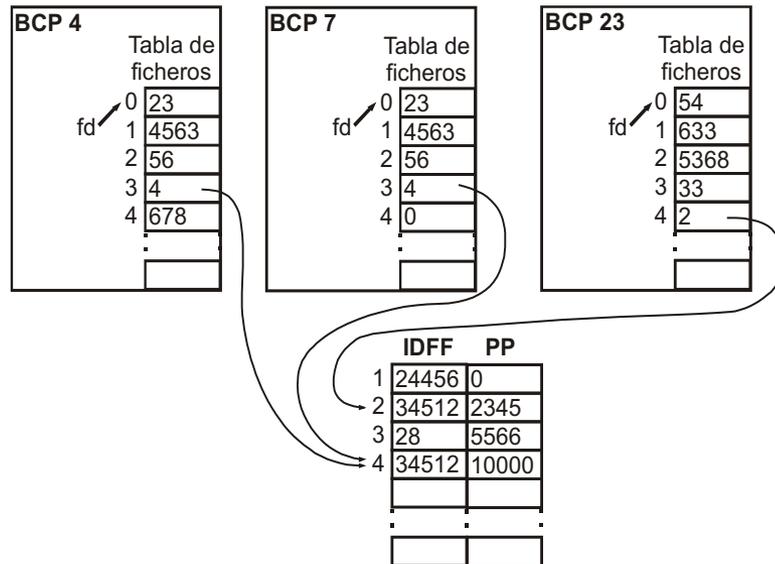
- ▶ La información de cada proceso está en el BCP...
- ▶ Información fuera del BCP:
 - ▶ Por razones de eficiencia
 - ▶ Para compartir información entre procesos



- ▶ Ejemplos:
 - ▶ Tabla de segmentos y páginas de **memoria**
 - ▶ Tabla de punteros de posición de **ficheros**
 - ▶ Lista de peticiones a **dispositivos**

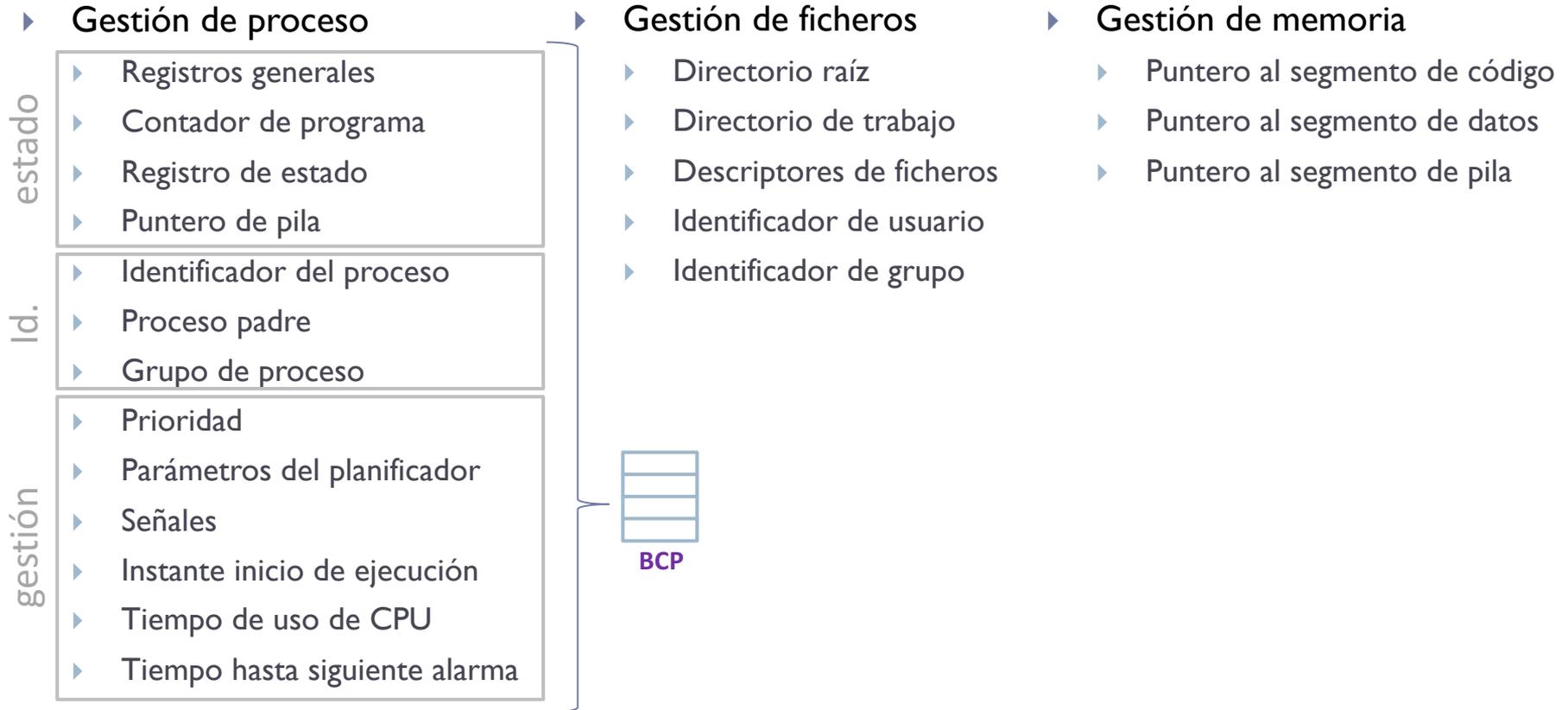
Dónde: información del proceso

- ▶ Tabla de punteros de posición de ficheros:



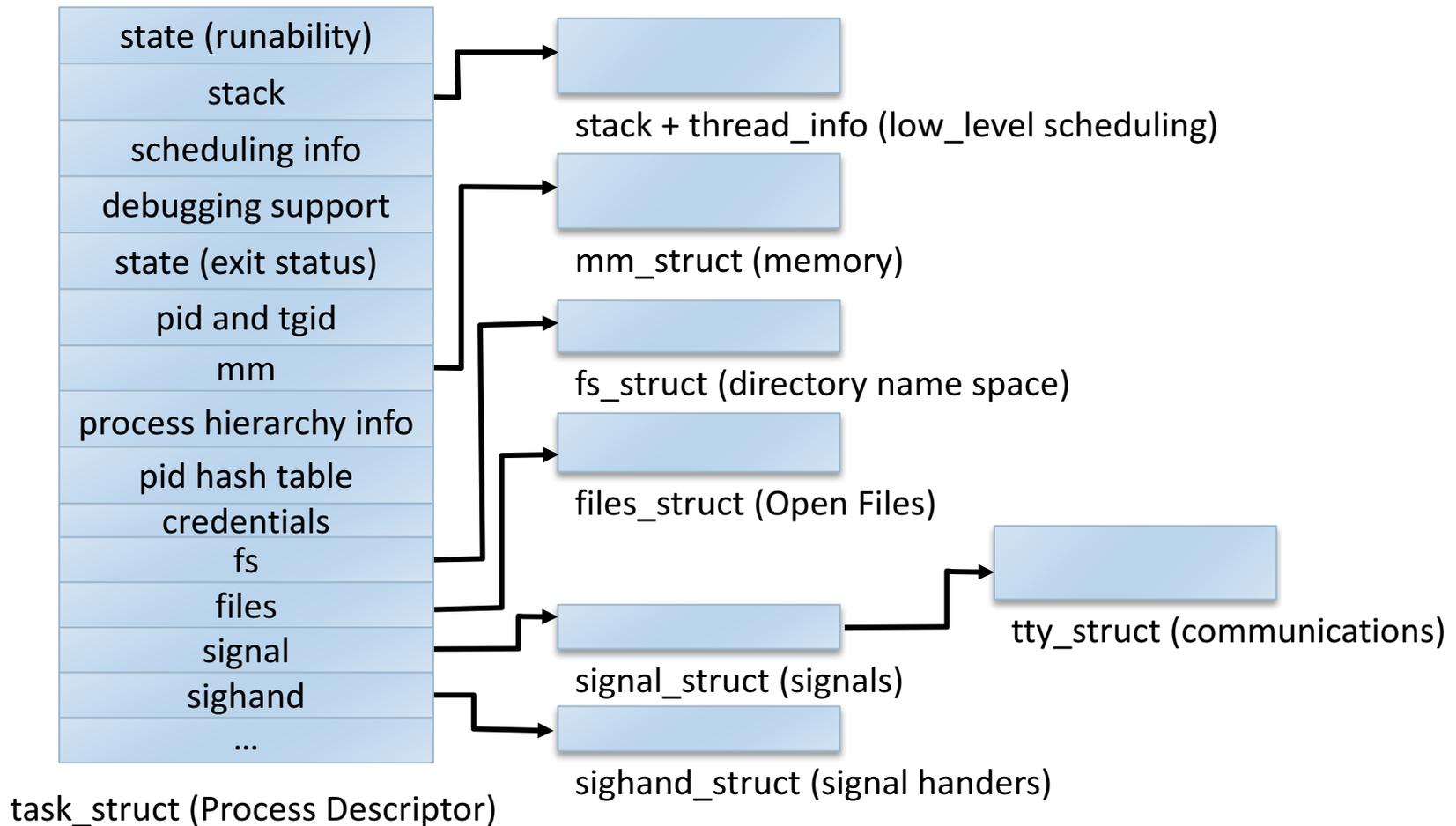
- ▶ Describe la posición de lectura/escritura de los ficheros abiertos
- ▶ La compartición de estado del fichero entre procesos obliga a que sea externa al BCP
- ▶ El BCP contiene el índice del elemento de la tabla que contiene la información del fichero abierto: el i-nodo y la posición de lectura/escritura.

Información del proceso



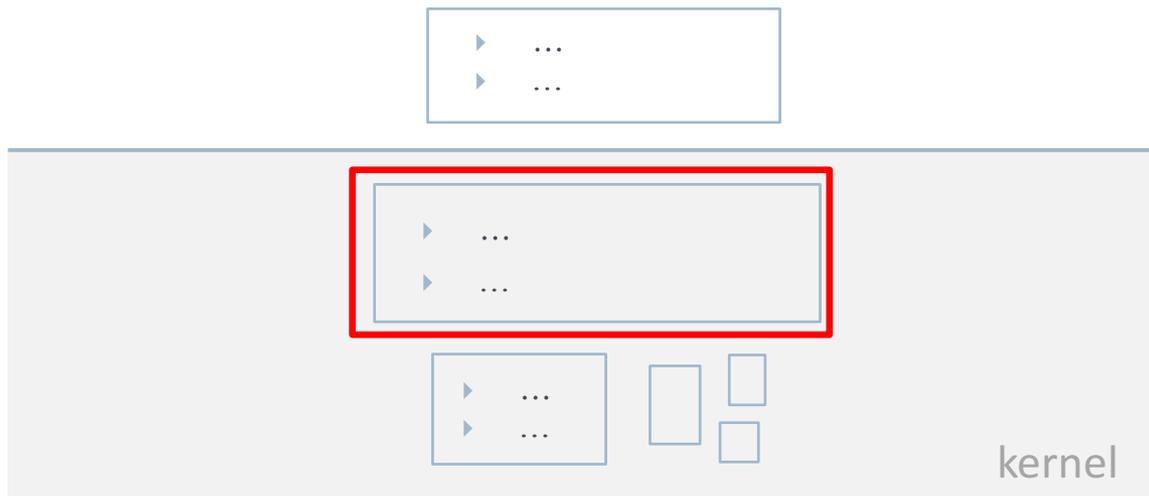
Información del proceso

Linux



Servicios del sistema operativo

inicialización y finalización de procesos



Creación de procesos

- ▶ Un proceso se crea:
 - ▶ Durante el arranque del sistema
 - ▶ Hilos del kernel + primer proceso (Ej.: init, swapper, etc.)
 - ▶ Cuando un proceso existe hace una llamada al sistema para crear otro:
 - ▶ Cuando el sistema operativo comienza un nuevo trabajo
 - ▶ Cuando un usuario arranca un nuevo programa
 - ▶ Cuando durante la ejecución de un programa se necesite

Finalización de procesos

▶ Un proceso termina:

▶ De forma voluntaria:

- ▶ Finalización normal
- ▶ Finalización con error

▶ De forma involuntaria:

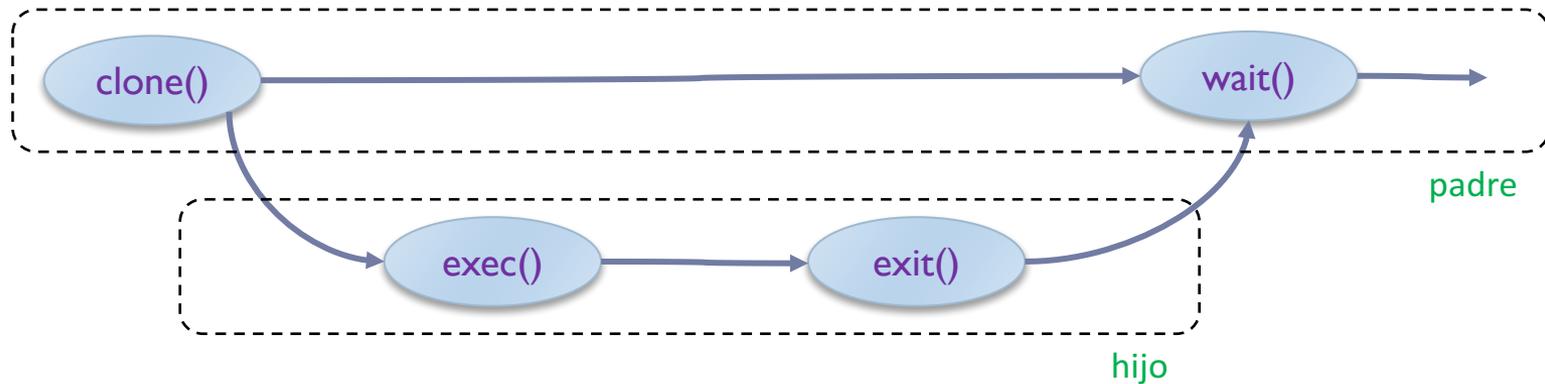
- ▶ Finalizado por el sistema (Ej.: excepción, sin recursos necesarios)
- ▶ Finalizado por otro proceso (Ej.: a través de llamada al sistema)
- ▶ Finalizado por el usuario (Ej.: control-c por teclado)

- ▶ En Unix/Linux se usan señales como mecanismo
- ▶ Se pueden capturar y tratar (salvo SIGKILL) para evitar finalizar involuntariamente

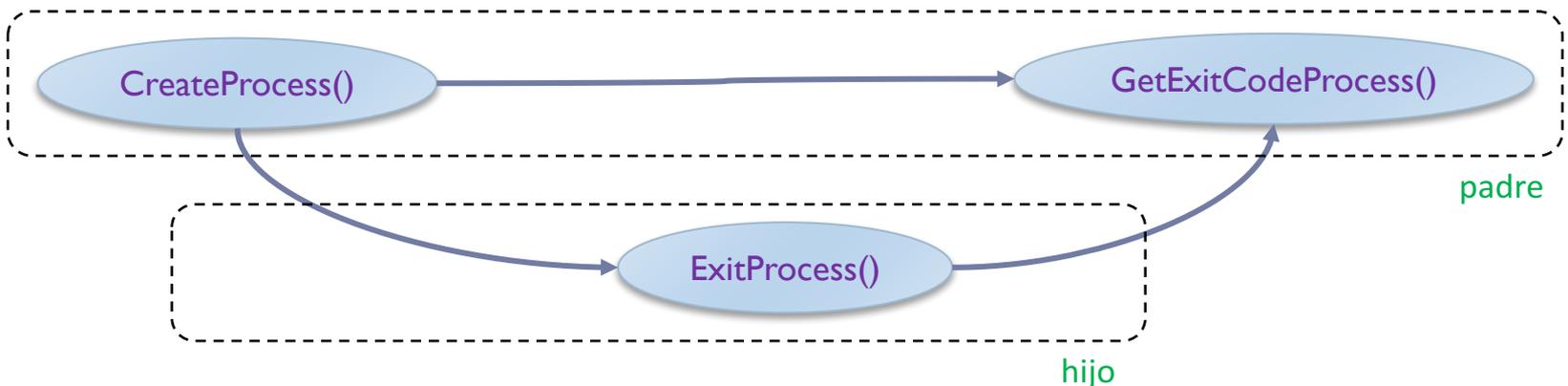
Creación y terminación de procesos

Llamadas al sistema

► Linux



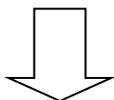
► Windows



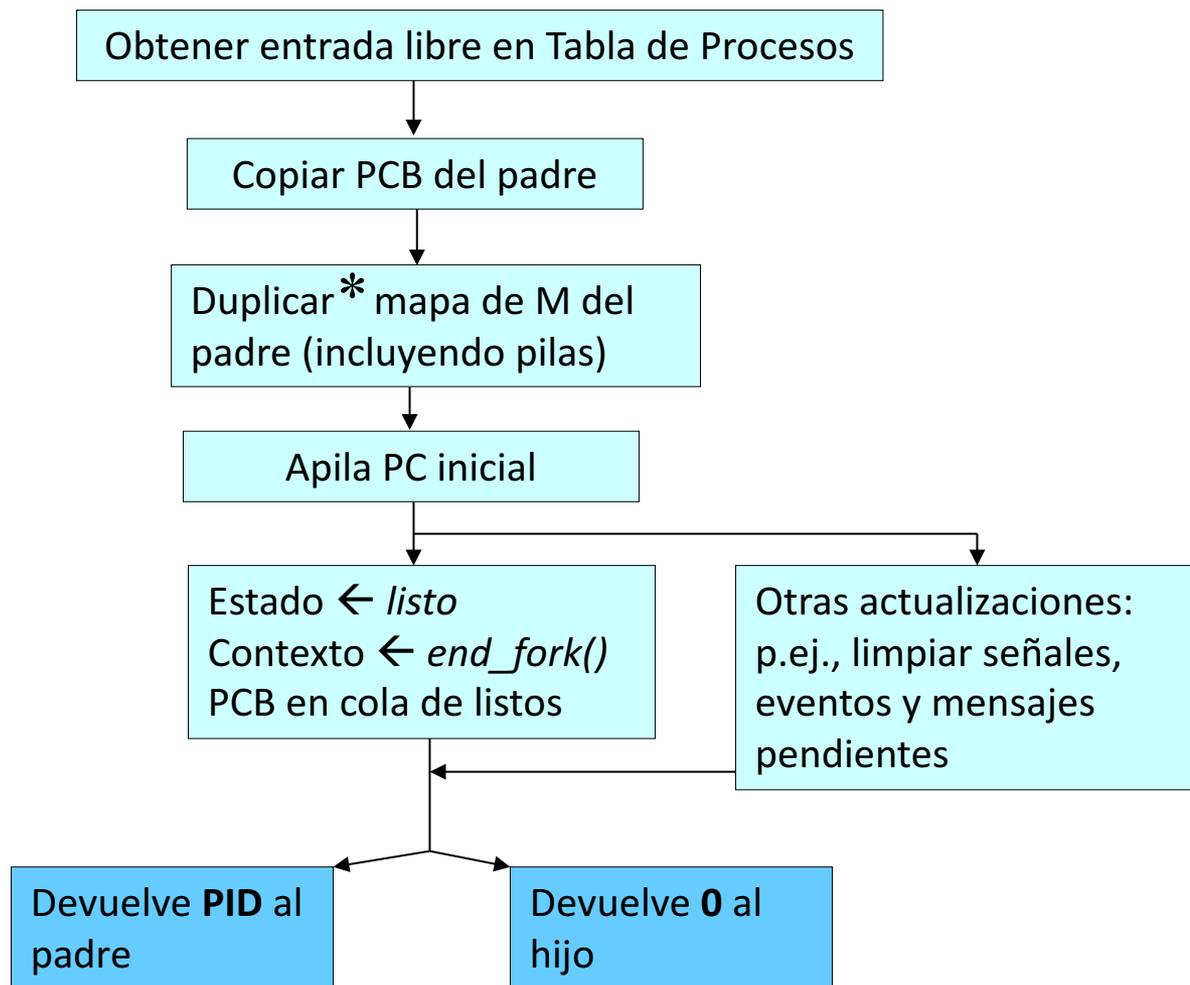
Creación de procesos

Linux: clone

clone:



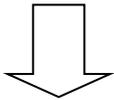
“Clona al proceso padre
y da una nueva
identidad al hijo”



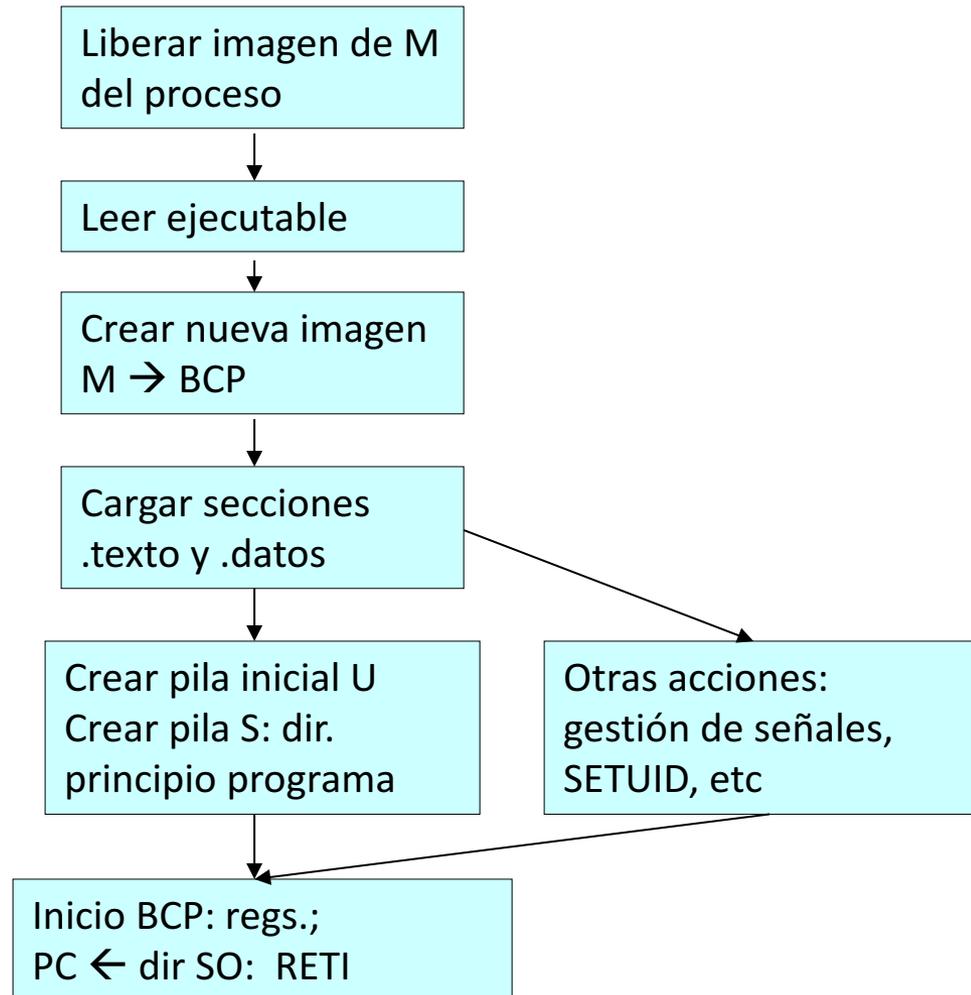
Cambio de imagen de un proceso

Linux: exec

exec:



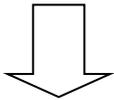
“Cambia la imagen de memoria de un proceso usando como ‘recipiente’ uno previo”



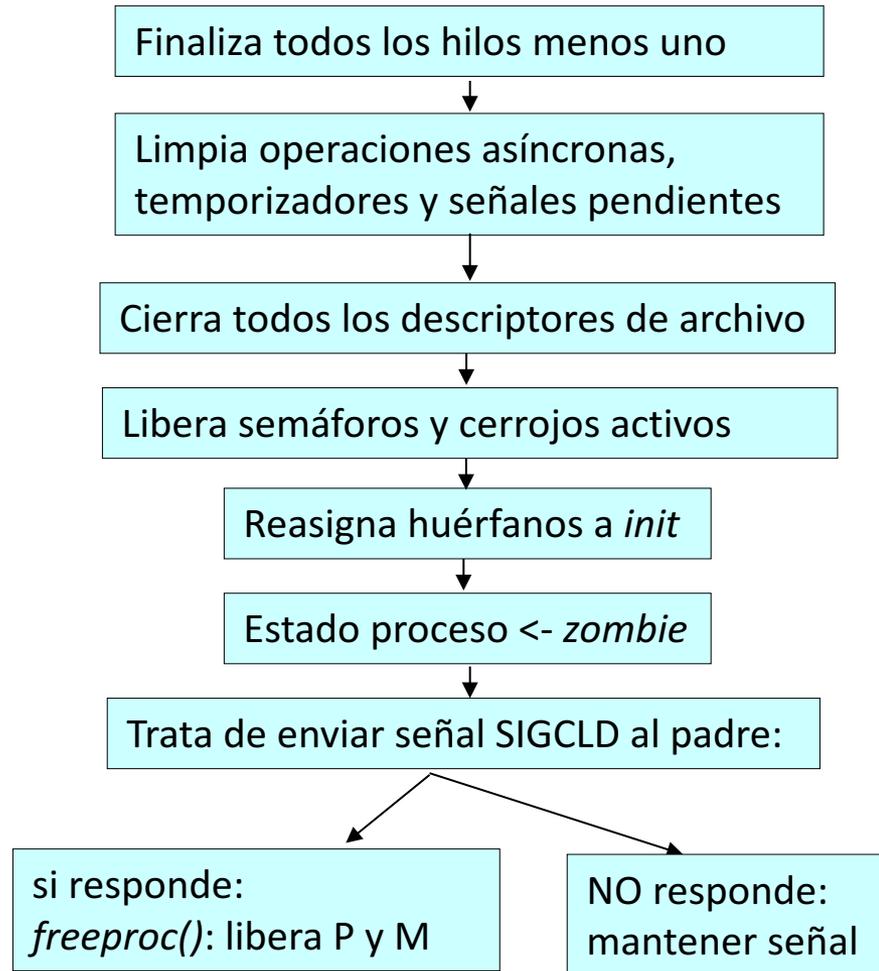
Terminación de procesos

Linux: exit

exit:

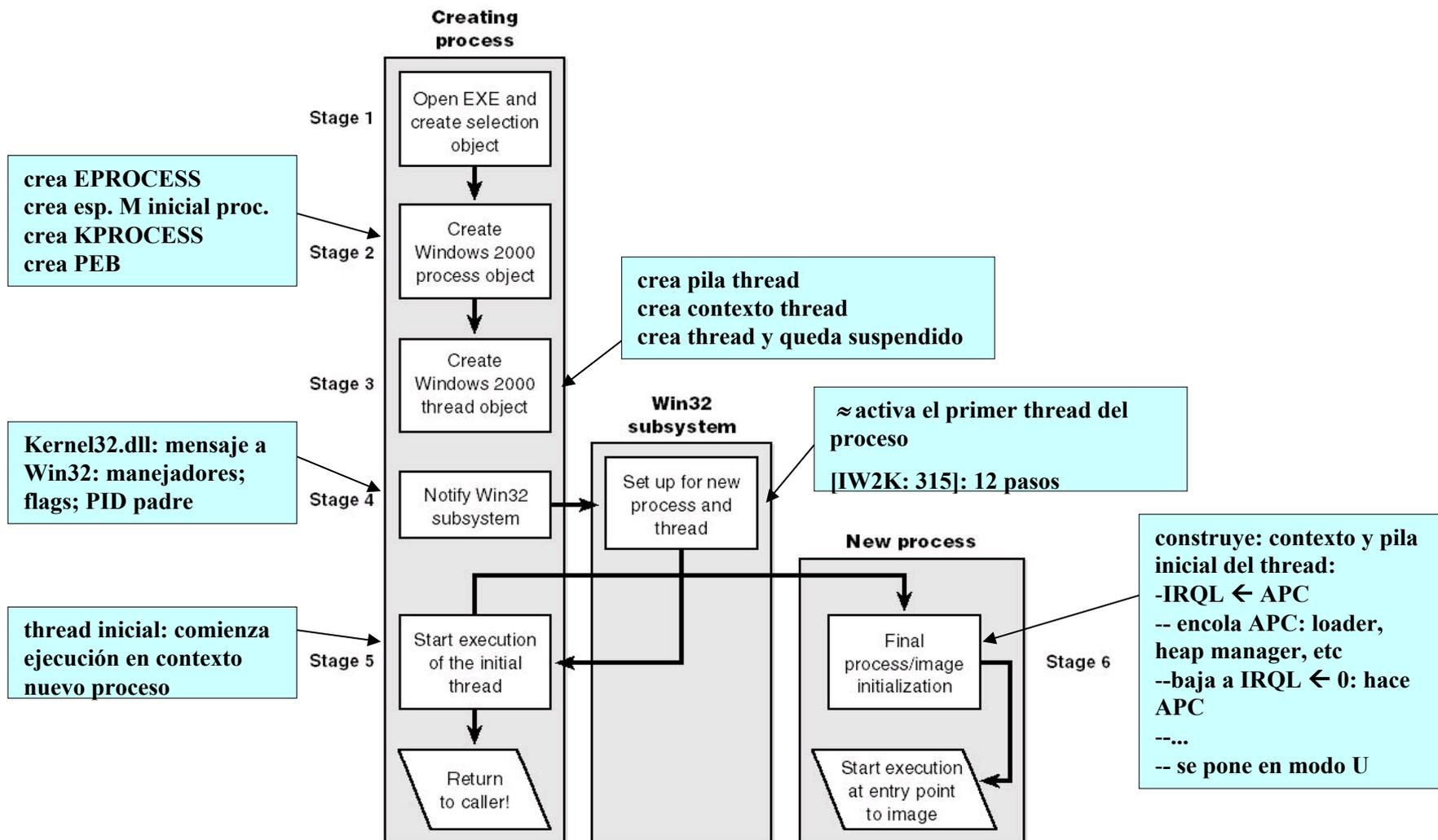


“Termina la ejecución de un proceso y libera los recursos”

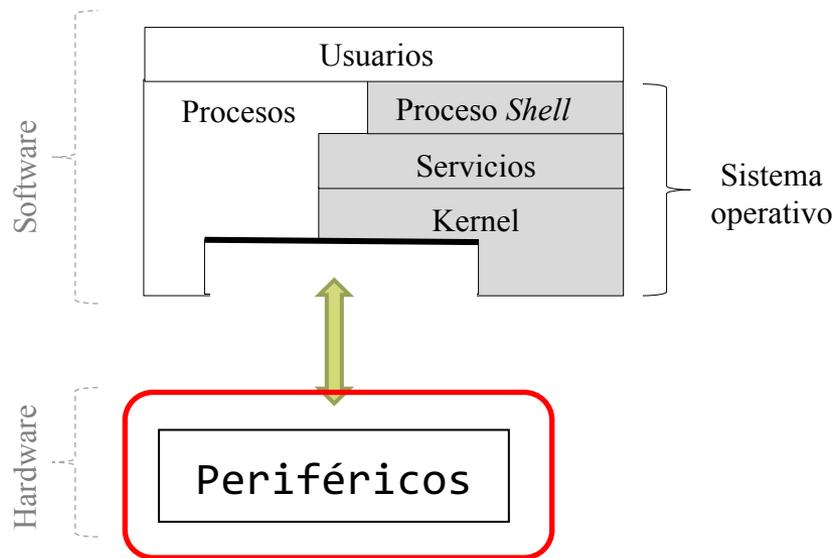


Creación de procesos

Windows: CreateProcess



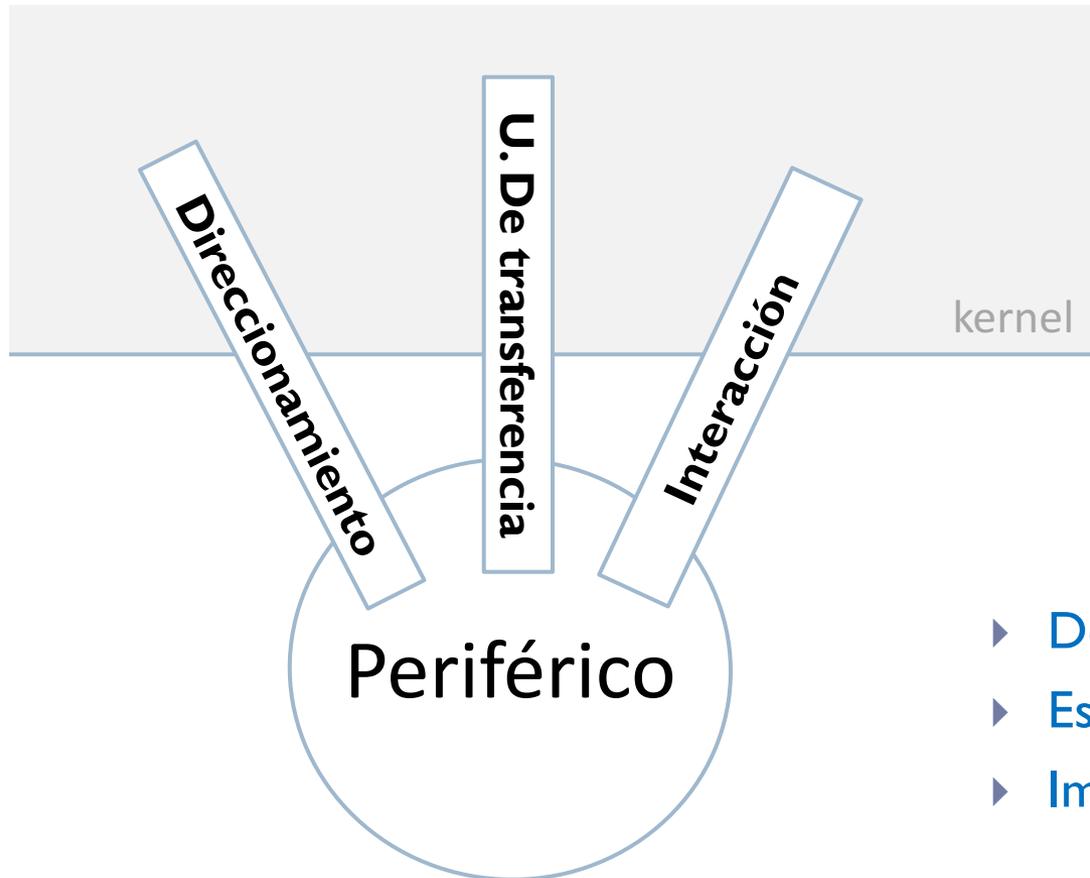
Contenidos



► **Procesos**

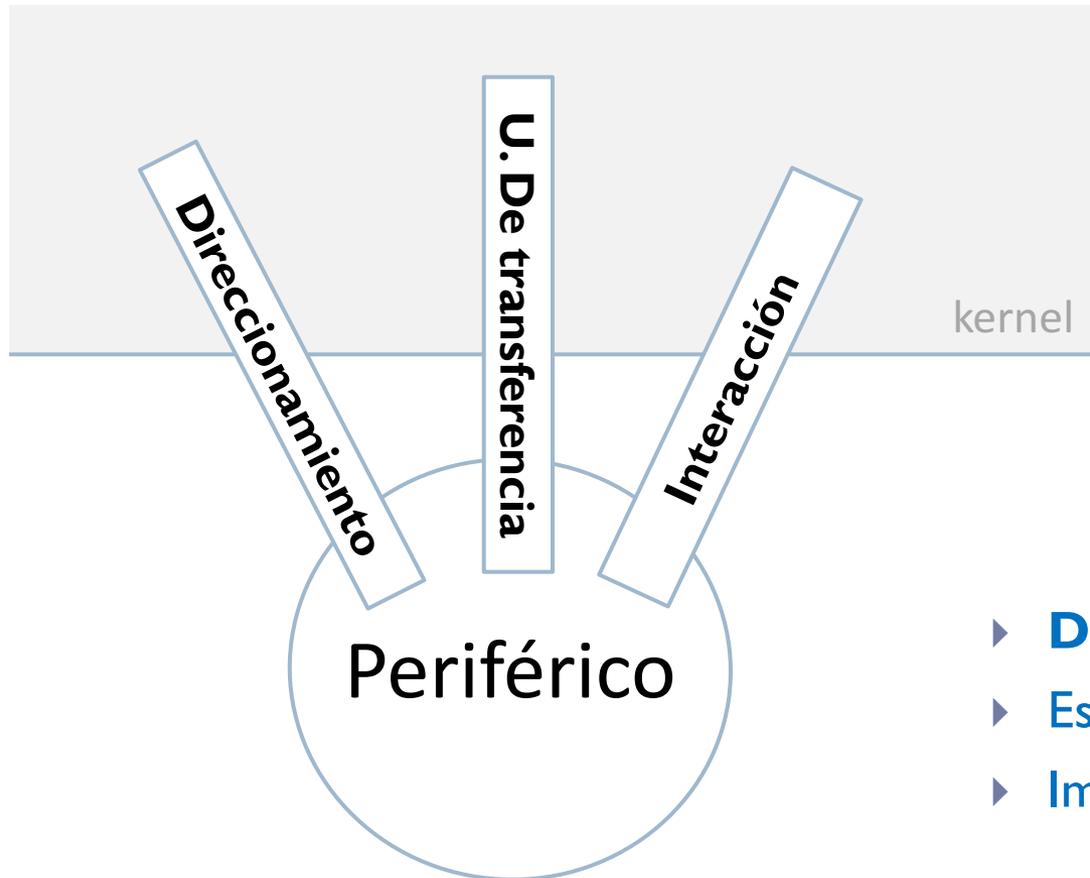
► **Periféricos**

Introducción



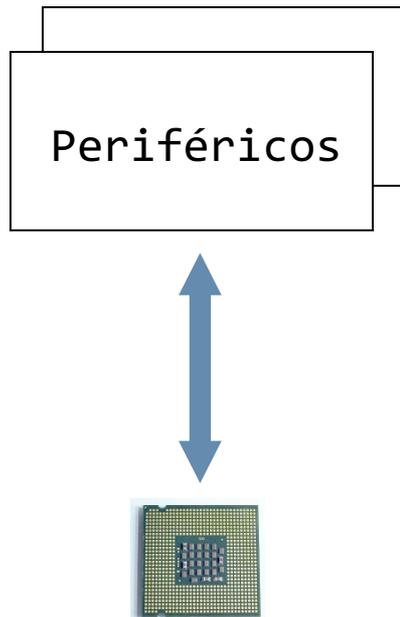
- ▶ Definición de periférico
- ▶ Estructura general
- ▶ Implicaciones en S.O.

Introducción



- ▶ **Definición de periférico**
- ▶ Estructura general
- ▶ Implicaciones en S.O.

Concepto de periférico



▶ **Periférico:**

- ▶ Todo aquel **dispositivo externo** que se conecta a una CPU a través de la unidad o **módulo de entrada/salida (E/S)**.
- ▶ Permiten **almacenar información** o **comunicar** el computador con el mundo **exterior**.

Clasificación de periféricos (por uso)



► Comunicación:

► **Hombre** - máquina

- (Terminal) teclado, ratón, ...
- (Impresa) plotter, escáner, ...

► **Máquina** - máquina (Módem, ...)

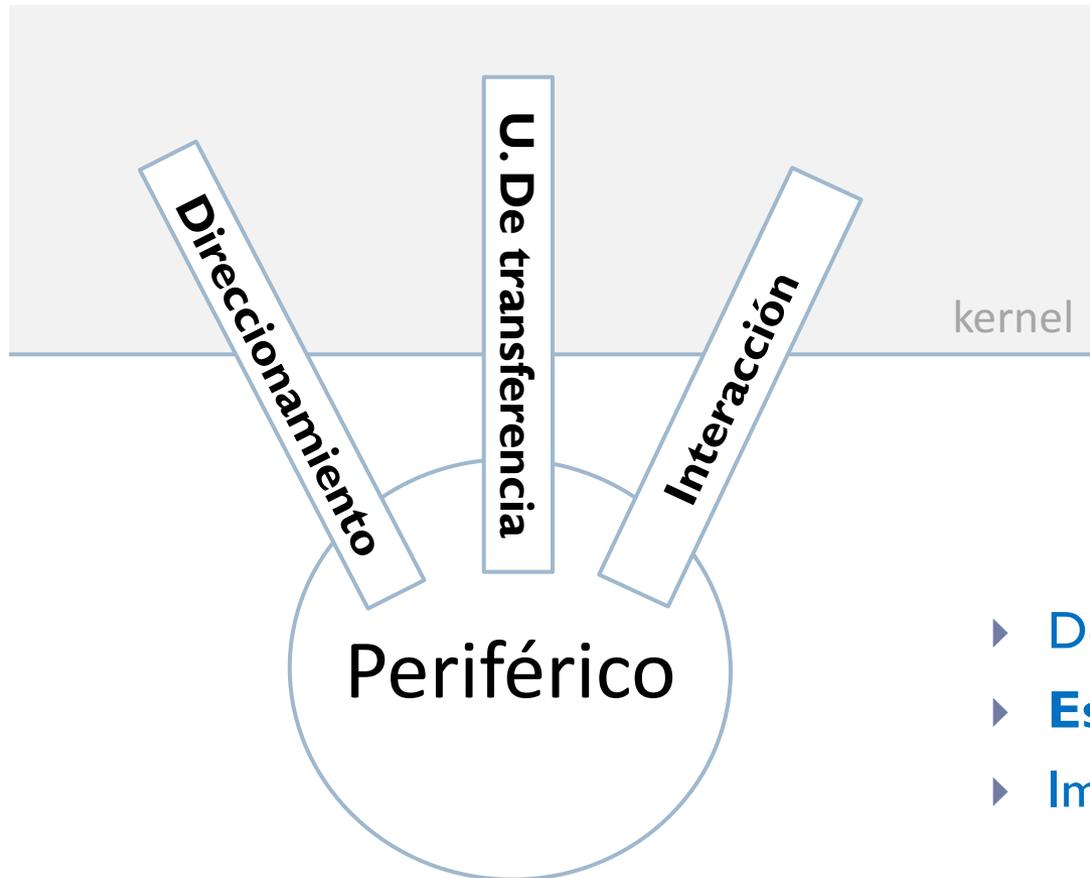
► **Medio físico** - máquina

- (Lectura/accionamiento) x (analógico/digital)

► Almacenamiento:

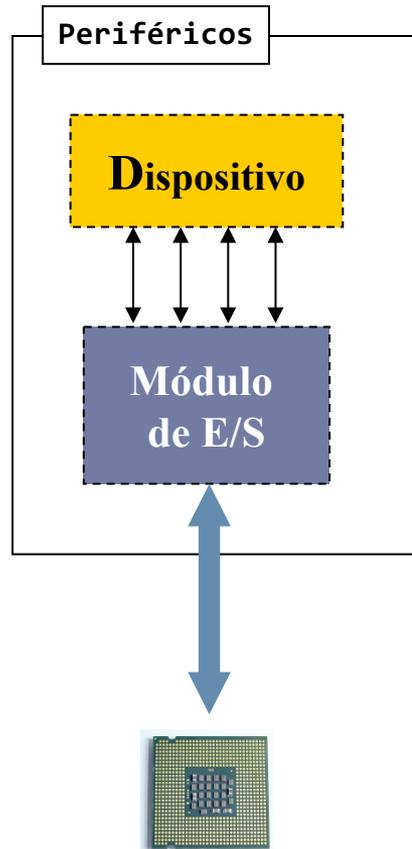
- Acceso “directo” (Discos, DVD, ...)
- Acceso secuencial (Cintas)

Introducción



- ▶ Definición de periférico
- ▶ **Estructura general**
- ▶ Implicaciones en S.O.

Estructura general de un periférico

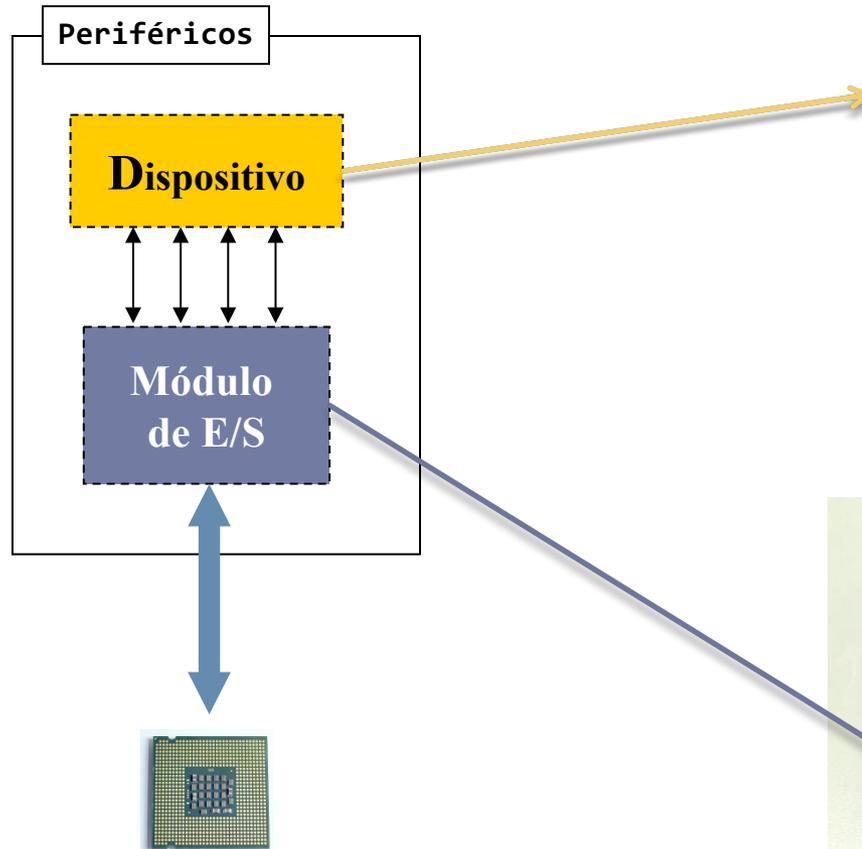


- ▶ Compuesto de:
 - ▶ **Dispositivo**
 - ▶ Hardware que interactúa con el entorno
 - ▶ **Módulo de Entrada/Salida**
 - ▶ También denominado **controlador**
 - ▶ Interfaz entre dispositivo y la CPU, que le oculta las particularidades de este

Periférico = Dispositivo + Controlador

Ejemplo

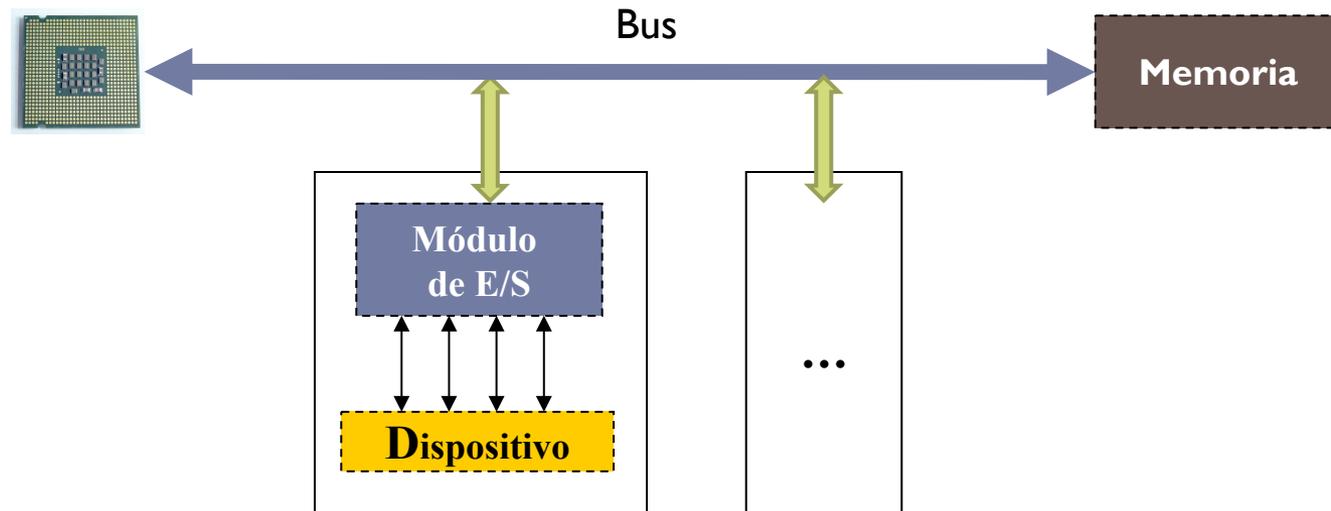
Disco duro



Módulo de E/S

qué son

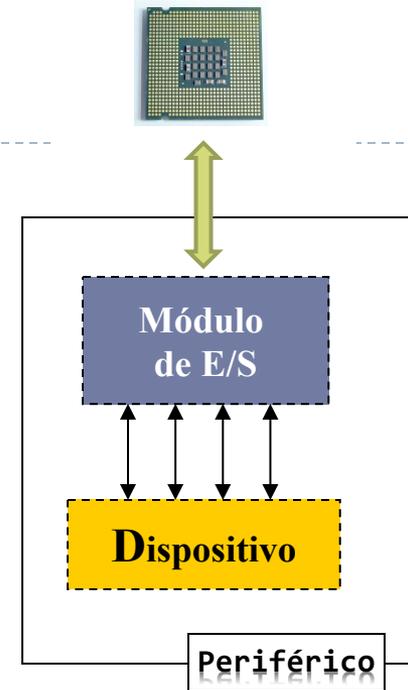
- ▶ Las **unidades o módulos de E/S** realizan la conexión de la CPU con los dispositivos periféricos.



Módulo de E/S

necesidad

- ▶ Son necesarios debido a:
 - ▶ Gran variedad de periféricos.
 - ▶ Los periféricos son 'raros'
 - ▶ La velocidad de transferencia de datos de los periféricos es mucho menor que la de la memoria o el procesador.
 - ▶ Los periféricos son 'muy lentos'
 - ▶ Formatos y tamaños de palabra de los periféricos distintos a los del computador al que se conectan.

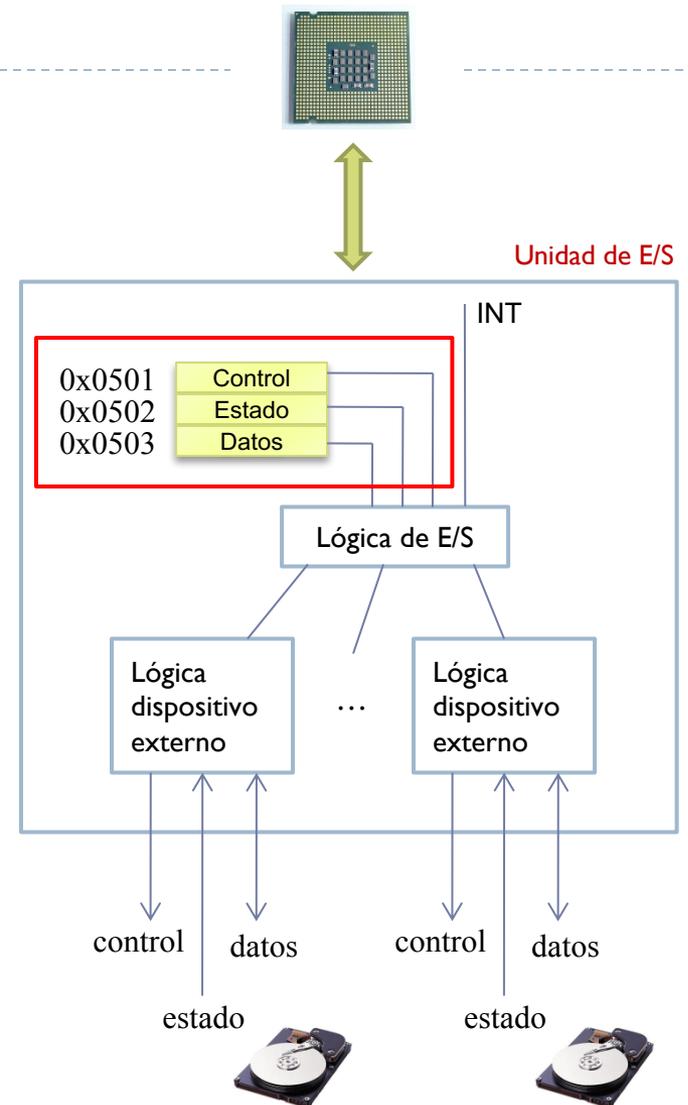


Módulo de E/S

estructura

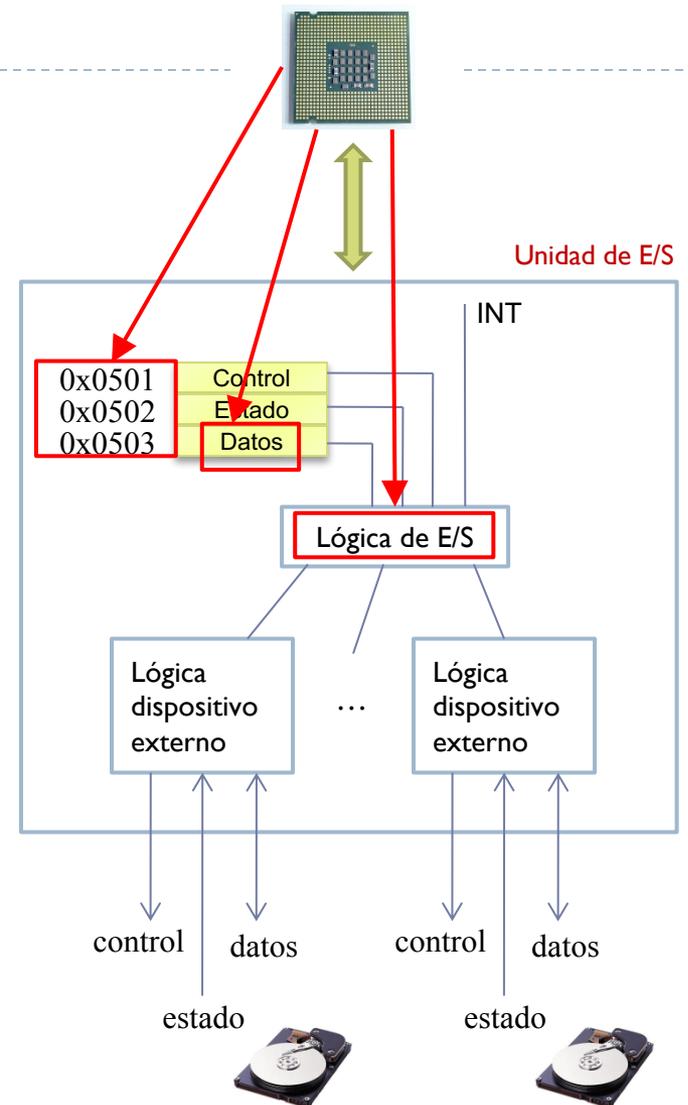
Interacción entre CPU y
Unidad de E/S a través de:

- ▶ **3 tipos** de registros:
 - ▶ Registro de **control**
 - ▶ Ordenes para el periférico
 - ▶ Registro de **estado**
 - ▶ Estado desde de la última orden
 - ▶ Registro de **datos**
 - ▶ Datos intercambiados CPU/periférico
- ▶ **1 tipo** de línea de interrupción:
 - ▶ Interrupción de **aviso**



Módulo de E/S funcionamiento

- ▶ Aspectos importantes:
 - ▶ Direccionamiento:
 - ▶ Conjunto, separado
 - ▶ Unidad de transferencia:
 - ▶ Carácter, bloque
 - ▶ Interacción
computador-controlador:
 - ▶ Programada, Interrupciones, DMA



| | |
|--------|---------|
| 0x0501 | Control |
| 0x0502 | Estado |
| 0x0503 | Datos |

(1 / 3) Direccionamiento de E/S

▶ Espacio de memoria conjunto

- ▶ Los registros del 'controlador' se proyectan en memoria y usando un conjunto de direcciones de memoria se acceden a dichos registros.

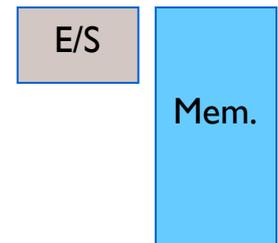
▶ Ej: `int * rctrl = 0x105A ;`
`(*rctrl) = 1 ;`



▶ Espacio de memoria separado (puertos)

- ▶ Con instrucciones ensamblador especiales (In/Out) se acceden a las direcciones de E/S (denominadas puertos) que representan los registros del 'controlador'.

▶ Ej: `out(0x105A, 1) ;`



| | |
|--------|---------|
| 0x0501 | Control |
| 0x0502 | Estado |
| 0x0503 | Datos |

(2/3) Unidad de transferencia

▶ **Dispositivos de bloque:**

- ▶ Unidad: **bloque** de bytes
- ▶ Acceso: **secuencial** o **directo**
- ▶ Operaciones: leer, escribir, situarse, ...
- ▶ Ejemplos: “cintas” y discos

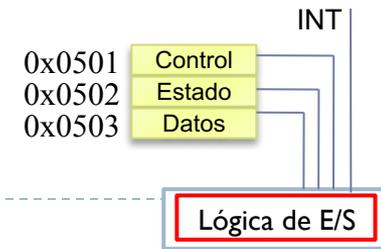


▶ **Dispositivos de carácter:**

- ▶ Unidad: **caracteres** (ASCII, Unicode, etc.)
- ▶ Acceso: **secuencial**
- ▶ Operaciones: get, put,
- ▶ Ejemplo: terminales, impresoras, etc.



(3/3) Interacción con computador



▶ E/S programada o directa

- ▶ CPU no hace otra cosa que E/S: espera → transfiere



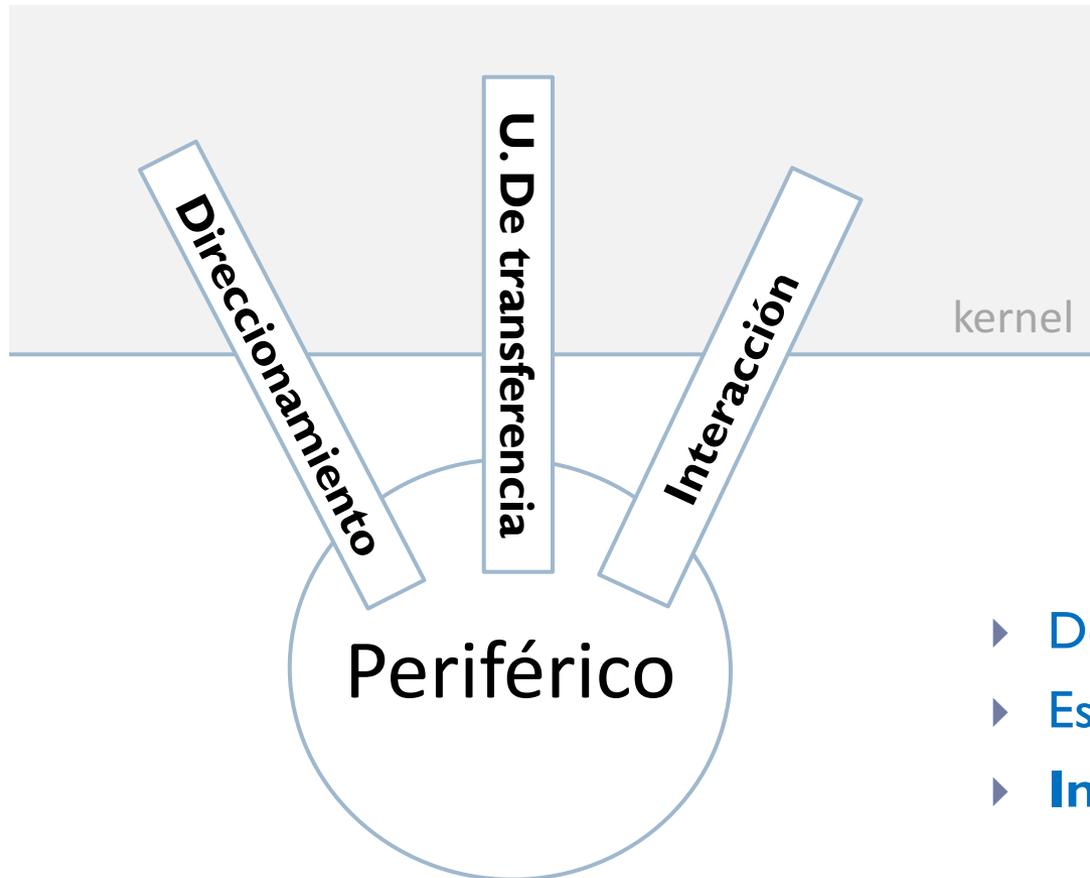
▶ E/S por interrupciones

- ▶ CPU no espera, sólo transfiere

▶ E/S por DMA (acceso directo a memoria)

- ▶ CPU no espera, no transfiere, le avisan del fin de los datos transferidos
 - Controlador de periférico más sofisticado (más costoso, mejor rendimiento)
 - Busca reducir la sobrecarga al transferir bloques de información

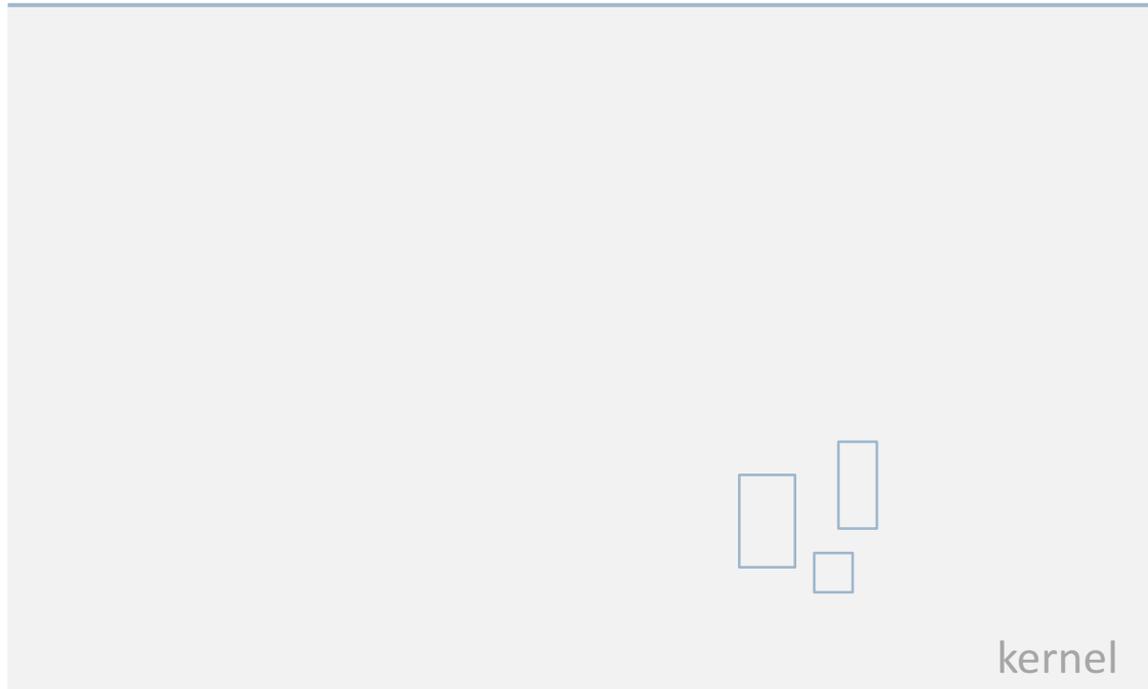
Introducción



- ▶ Definición de periférico
- ▶ Estructura general
- ▶ **Implicaciones en S.O.**

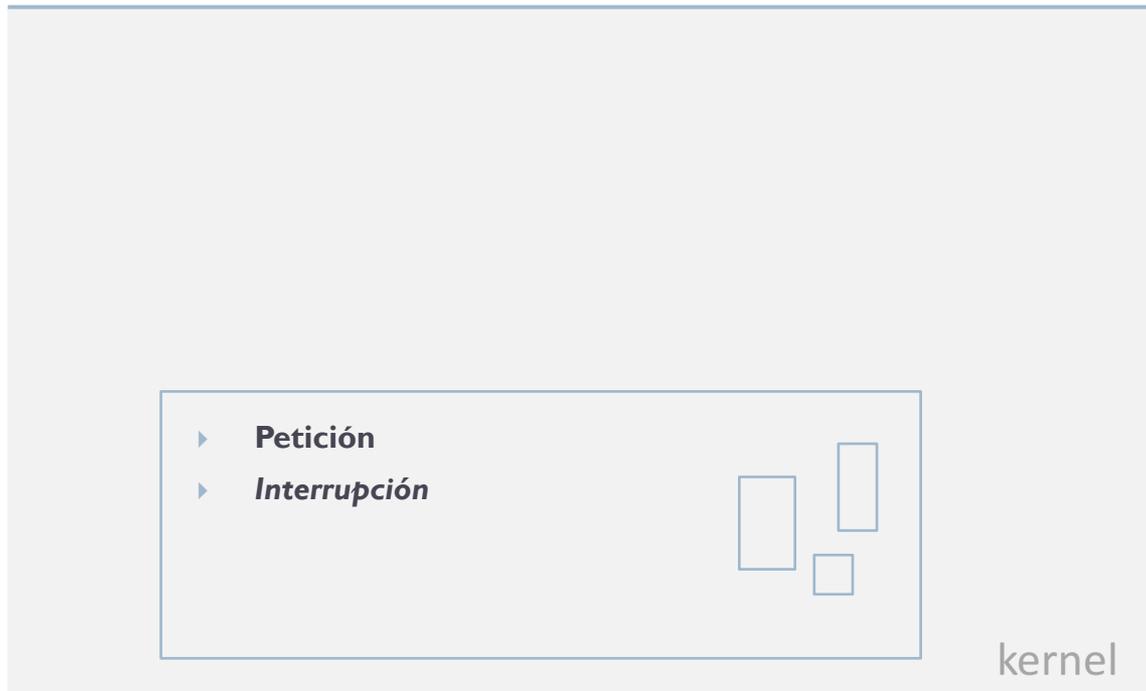
Implicaciones en el sistema operativo

I. Estructuras de datos



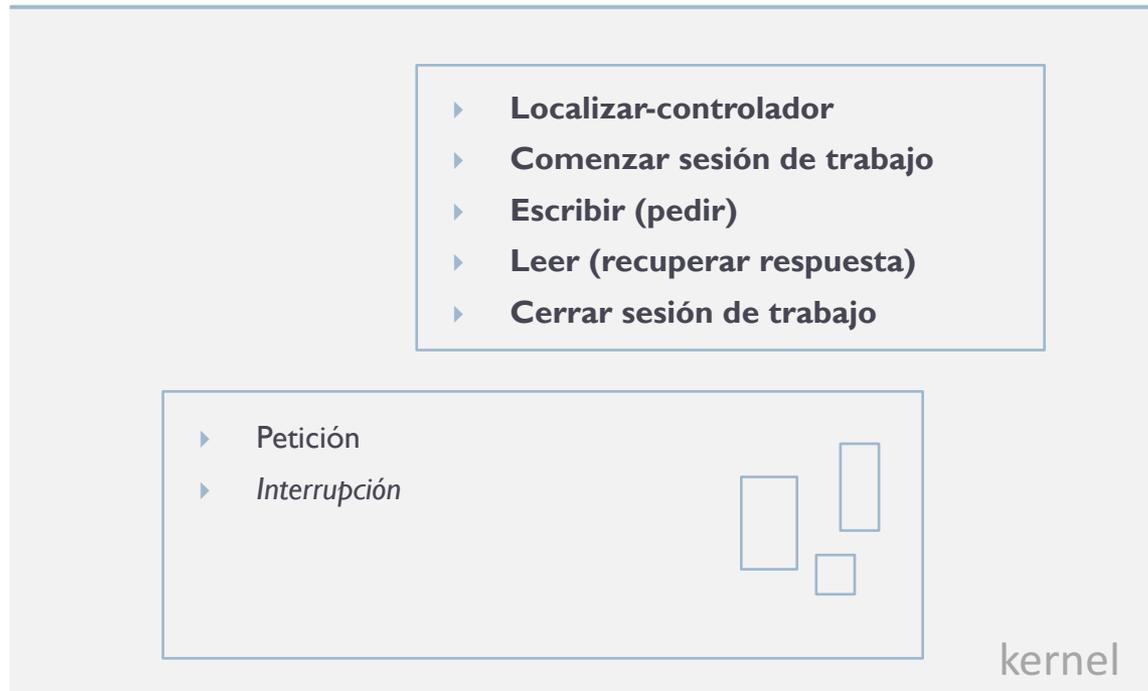
Implicaciones en el sistema operativo

2. Funciones: de gestión internas



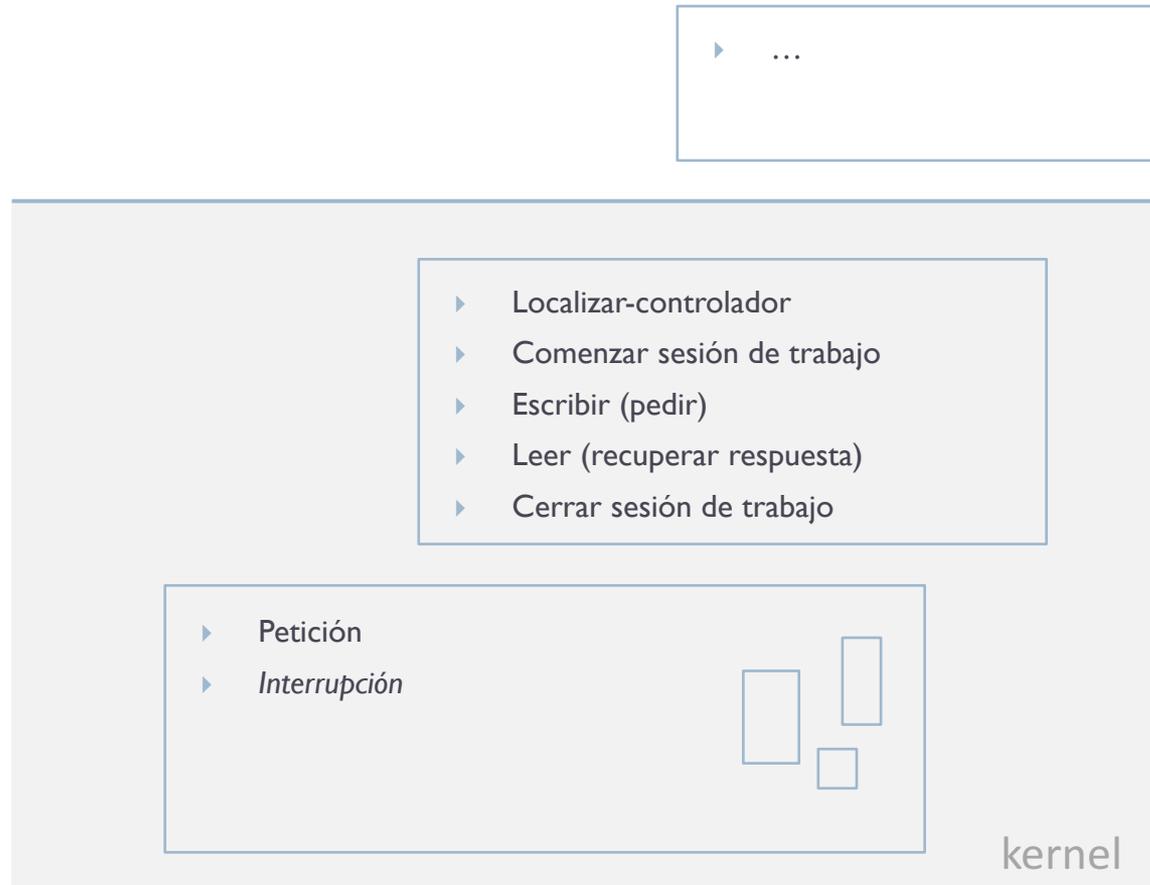
Implicaciones en el sistema operativo

3. Funciones: de servicio



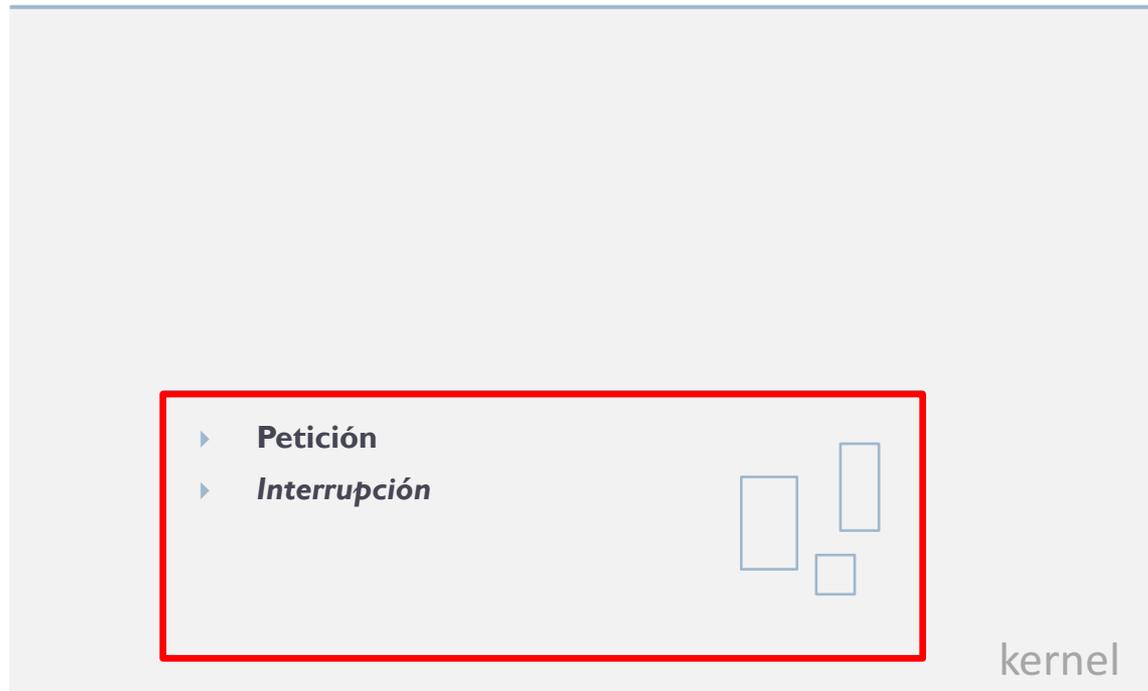
Implicaciones en el sistema operativo

3. Funciones: API de servicio



Implicaciones en el sistema operativo

(1 y 2) Estructuras de datos + funciones de gestión internas
= controlador (driver)



Principales tipos de protocolos

- ▶ **Petición -> respuesta individual**
 - ▶ Mayoría de dispositivos
- ▶ **Solo petición**
 - ▶ Ej.: tarjeta gráfica
 - ▶ E/S programada (rápidos o tiempo real)
- ▶ **Solo respuesta**
 - ▶ Ej.: reloj
 - ▶ E/S por interrupciones (generan datos sin petición previa)
- ▶ **Petición -> respuesta compartida**
 - ▶ Ej.: disco duro

Lección 3a

procesos, periférico, *drivers* y servicios ampliados

Diseño de Sistemas Operativos
Grado en Ingeniería Informática