

# Descripción de procesos y control

# Principales requisitos de un sistema operativo

- Intercalar la ejecución de varios procesos para maximizar el uso del procesador y a la vez proveer de un tiempo de respuesta razonable
- Asignar recursos a los procesos
- Soportar comunicación interprocesos y creación de procesos por el usuario

# Proceso

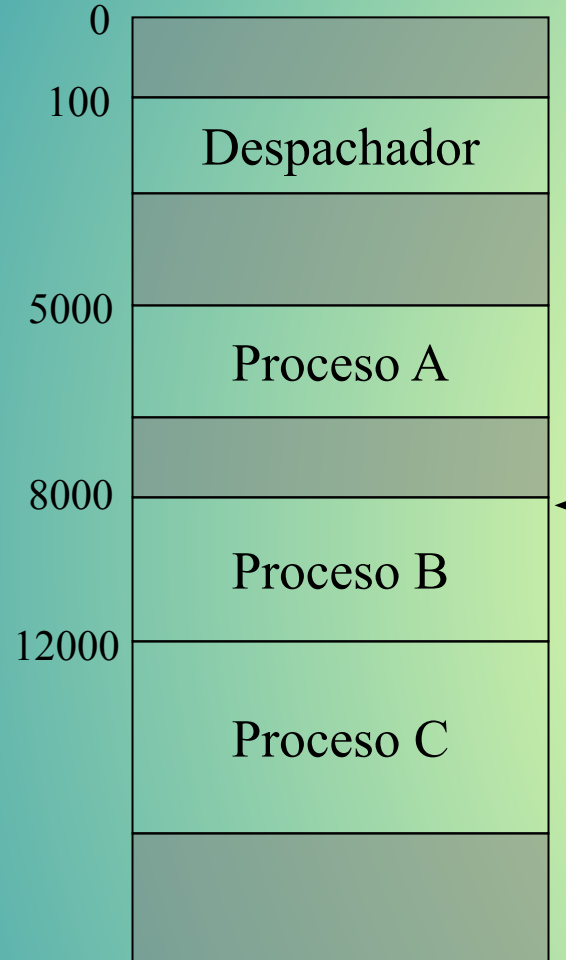
- Programa en ejecución
- Unidad de trabajo, también llamado tarea
- Entidad activa (programa es una entidad pasiva)

# Estados de un proceso

# Proceso

- Se puede rastrear
  - listar la secuencia de instrucciones que se ejecuta
- Está compuesto por
  - código del programa
  - actividad actual
  - pila
  - datos

# Ejemplo de ejecución. Procesos



Memoria Principal

**Despachador:** Programa que cambia el procesador de un proceso a otro

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.

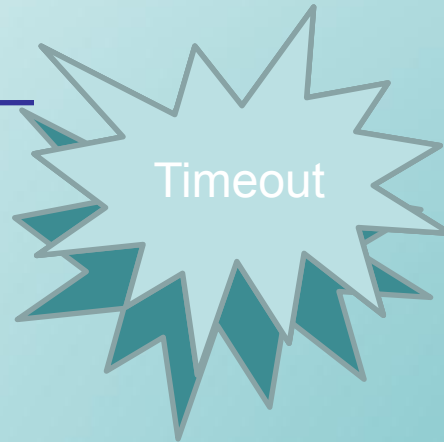
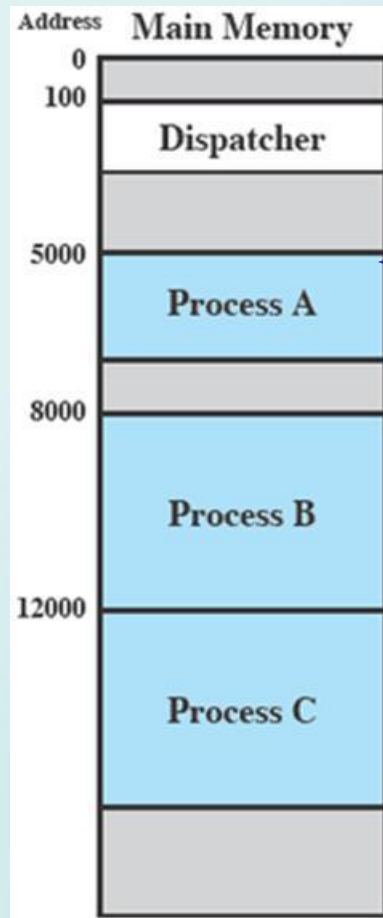
8000 = Dirección de comienzo del programa del Proceso B.

12000 = Dirección de comienzo del programa del Proceso C.

**Figura 3.3.** Traza de los procesos de la Figura 3.2.



# Traza del procesador



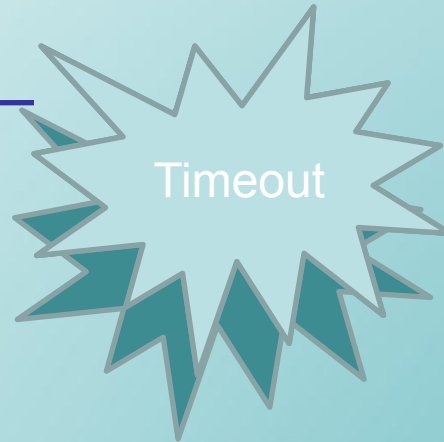
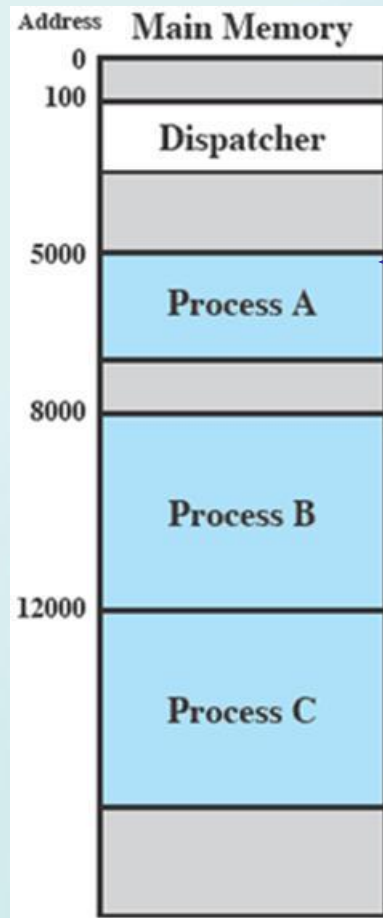
1	5000		
2	5001		
3	5002		
4	5003		
5	5004		
6	5005		
		Timeout	
7	100		
8	101		
9	102		
10	103		
11	104		
12	105		
13	8000		
14	8001		
15	8002		
16	8003		
		I/O Request	
17	100		
18	101		
19	102		
20	103		
21	104		
22	105		
23	12000		
24	12001		
25	12002		
26	12003		
27	12004		
28	12005		
		Timeout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

100 = Starting address of dispatcher program

Shaded areas indicate execution of dispatcher process;  
first and third columns count instruction cycles;  
second and fourth columns show address of instruction being executed

Figure 3.4 Combined Trace of Processes of Figure 3.2

# Traza del procesador



1	5000	27	12004
2	5001	28	12005
3	5002	Timeout	
4	5003	29	100
5	5004	30	101
6	5005	31	102
Timeout		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	Timeout	
16	8003	41	100
I/O Request		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
		Timeout	

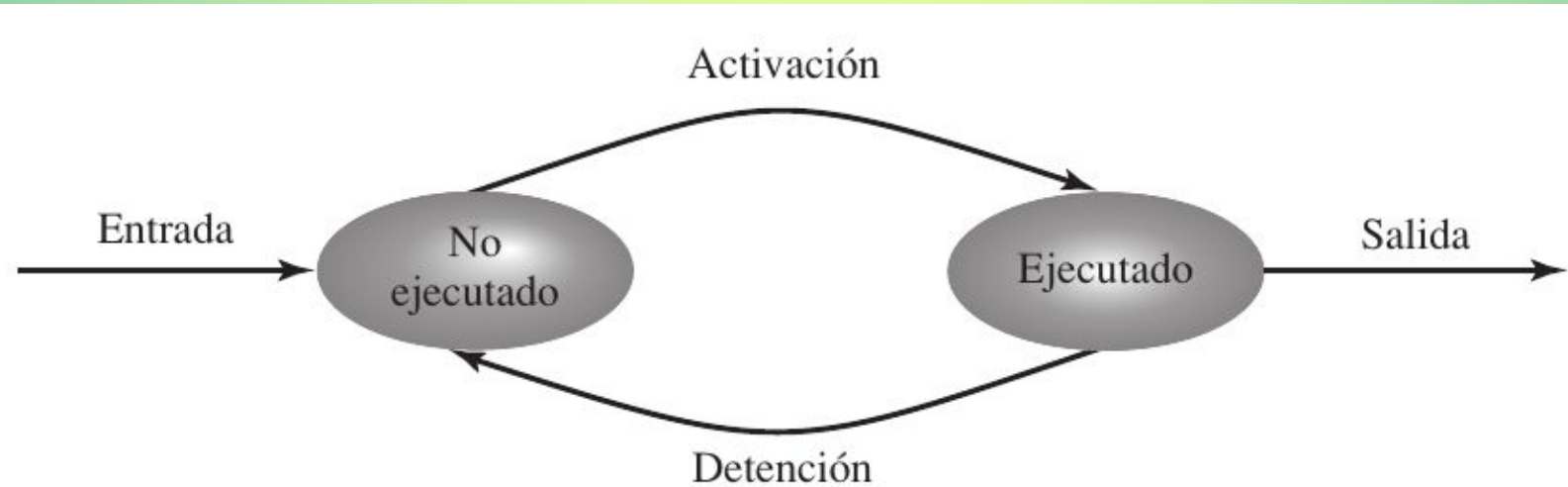
100 = Starting address of dispatcher program

Shaded areas indicate execution of dispatcher process;  
first and third columns count instruction cycles;  
second and fourth columns show address of instruction being executed

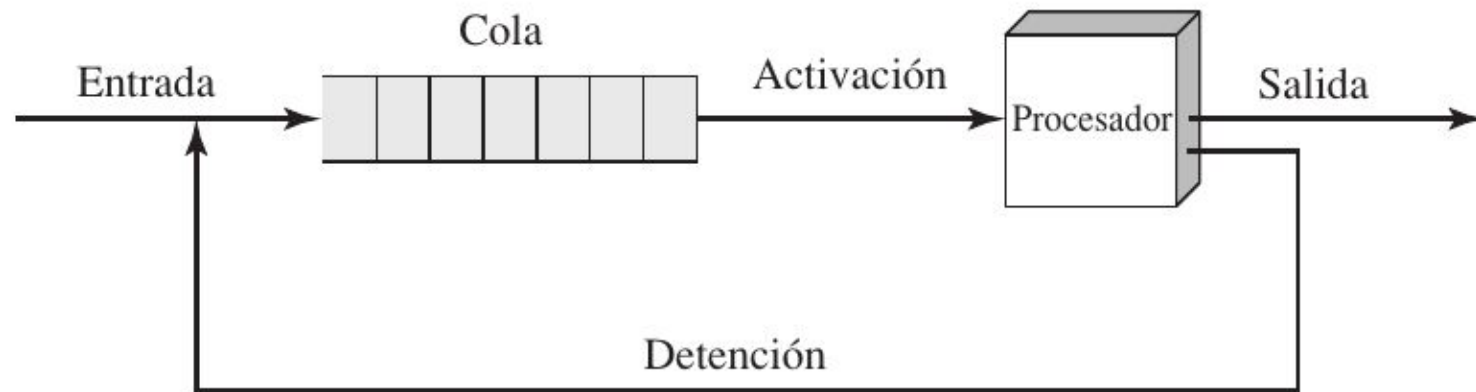
Figure 3.4 Combined Trace of Processes of Figure 3.2



# Modelo de dos estados de proceso



(a) Diagrama de transiciones de estados



(b) Modelos de colas

**Figura 3.5.** Modelo de proceso de dos estados.

# Creación de proceso

- Sumisión de una tarea por lotes
- Ingreso al sistema por parte de un usuario (log on)
- Creado por el SO para proveer un servicio
- Creado por otro proceso existente (spawning)

# Terminación de un proceso

- El trabajo de lotes realiza una operación de detención
- El usuario realiza un log off
- El proceso ejecuta un pedido de terminación del servicio
- Pedido de terminación (o terminación) del padre
- Condiciones de error y falla

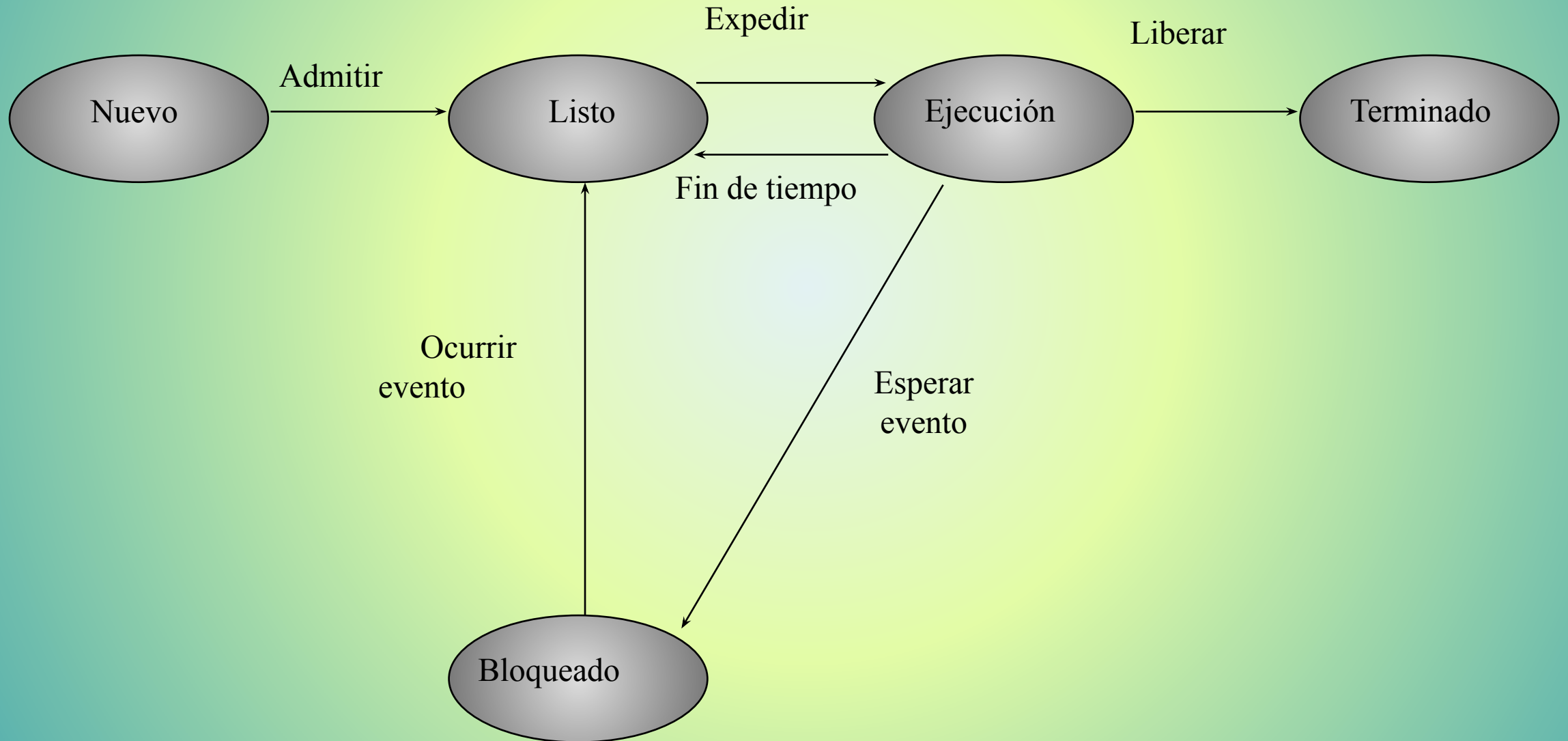
# Modelo de 5 estados de procesos

- Sin ejecutar
  - listo para ejecutar
  - Bloqueado
    - esperando E/S
- El despachador no puede simplemente seleccionar el proceso que ha esperado más tiempo porque puede estar bloqueado

# Modelo de cinco estados de proceso

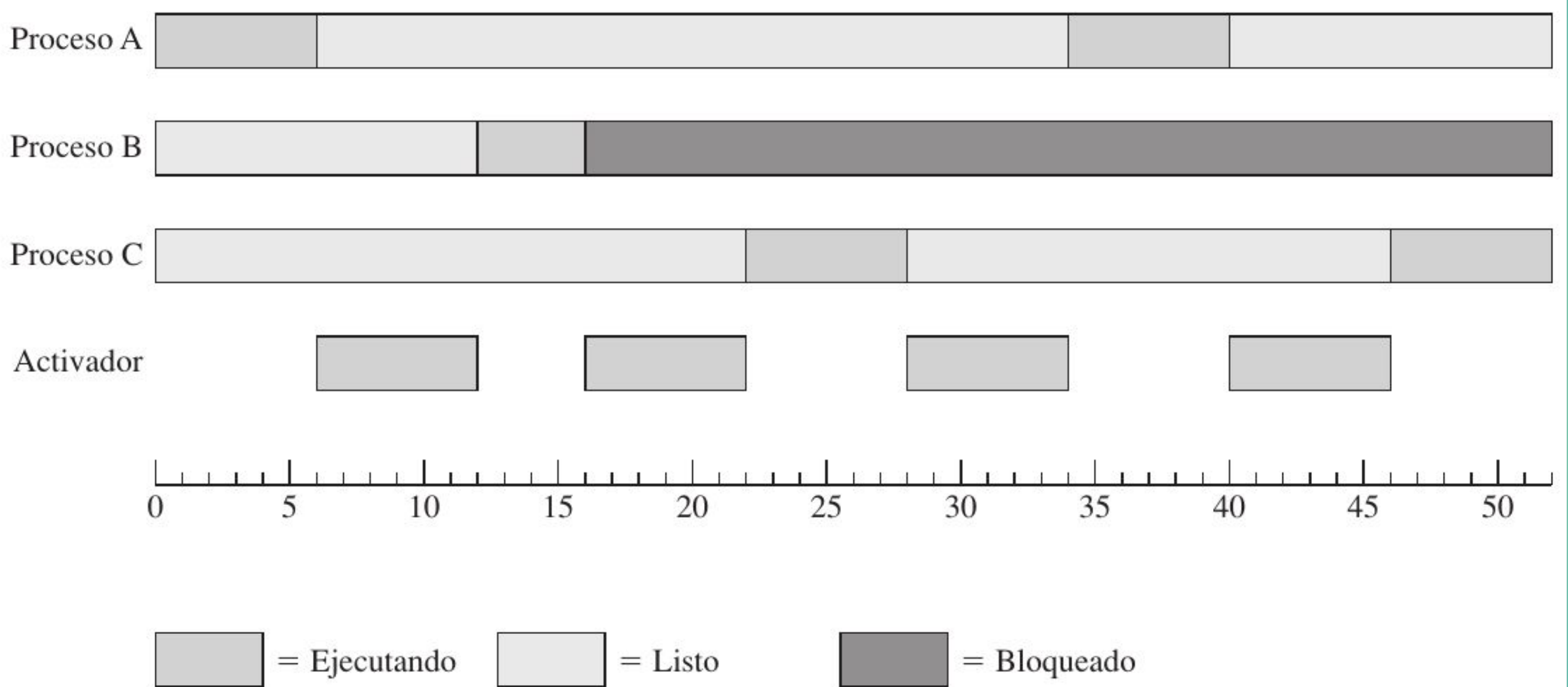
- Ejecución
- Listo
- Bloqueado
- Nuevo
- Terminado

# Modelo de cinco estados de proceso



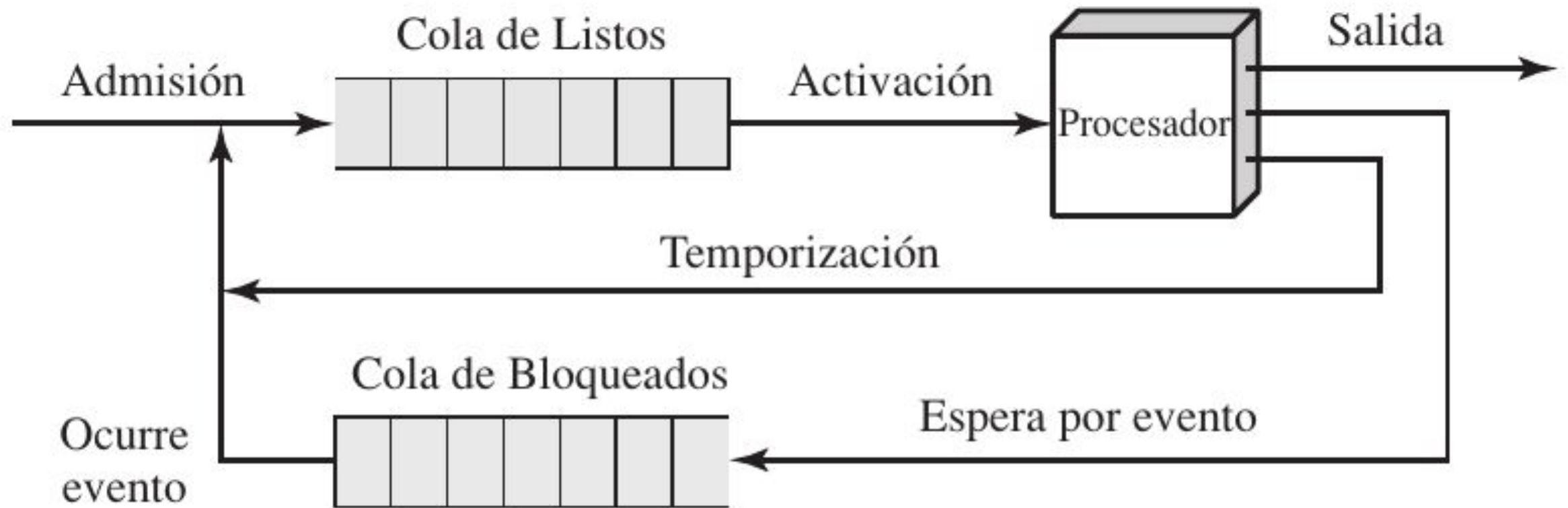


# Estado de Procesos



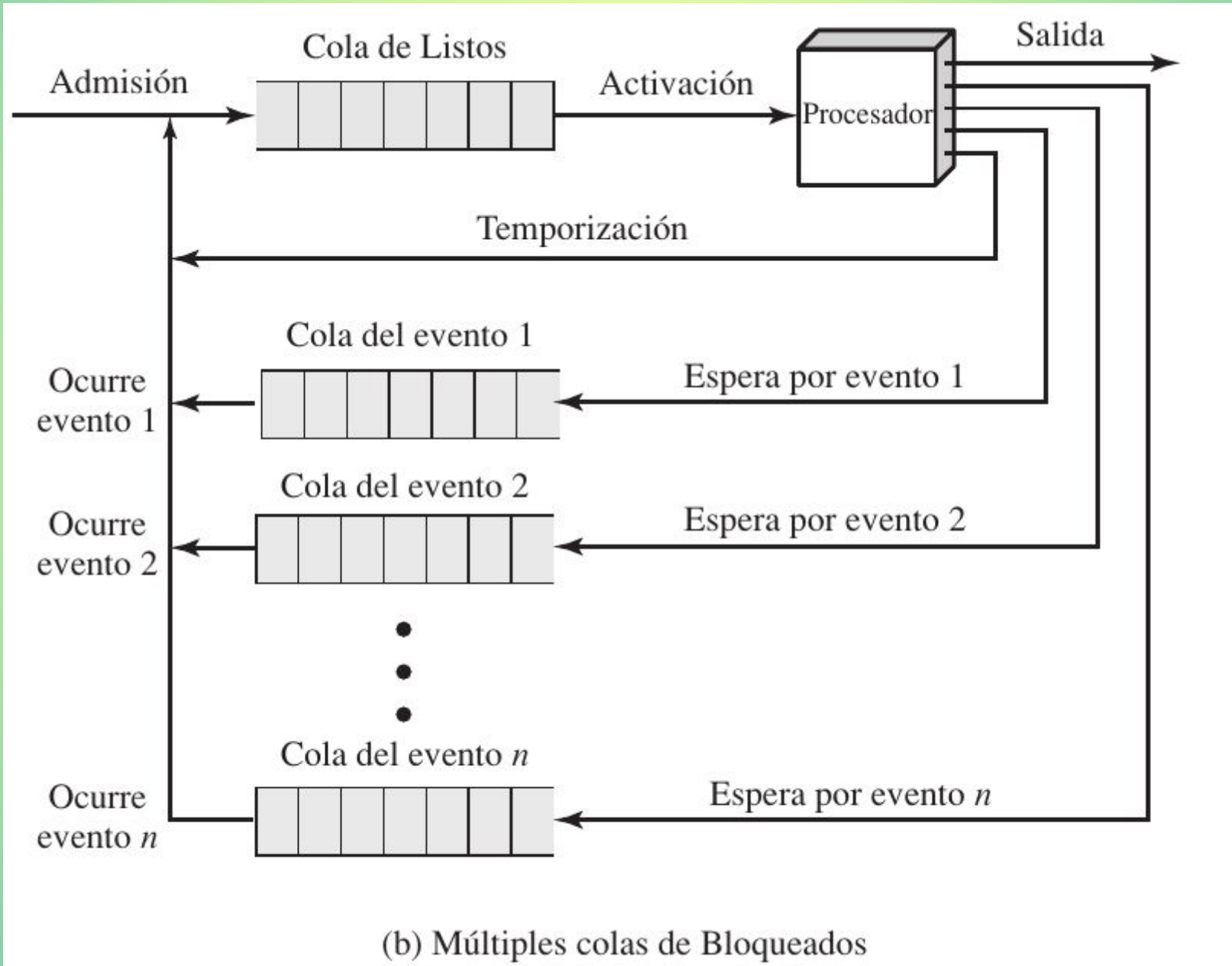
**Figura 3.7.** Estado de los procesos de la traza de la Figura 3.4.

# Cola simple de bloqueados



(a) Cola simple de Bloqueados

# Múltiples colas de bloqueados



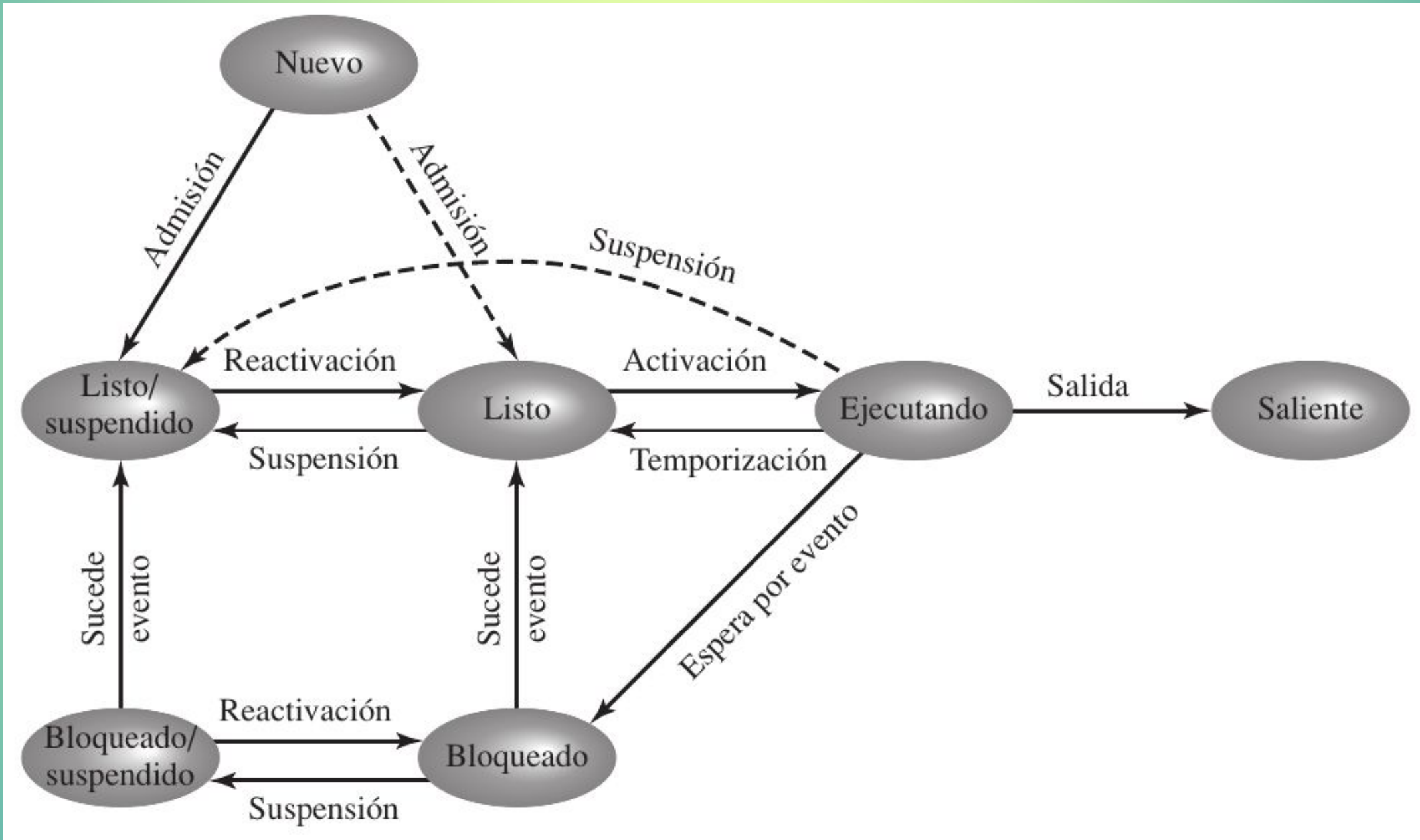
# Procesos suspendidos

- El procesador es más rápido que E/S entonces todos los procesos podrían estar esperando por E/S
- Intercambiar estos procesos con el disco para liberar más memoria
- Estado bloqueado se convierte en estado suspendido cuando se ha sacado a disco
- Dos nuevos estados
  - Bloqueado, suspendido
  - Listo, suspendido

# Razones para suspensión

- Swapping
- Otras razones del sistema operativo
  - el proceso está causando problemas
- Pedido de un usuario interactivo
- Cronómetro
  - proceso que se ejecuta a intervalos
- Pedido del proceso padre

# Diagrama de transición de estado de procesos con dos suspendidos



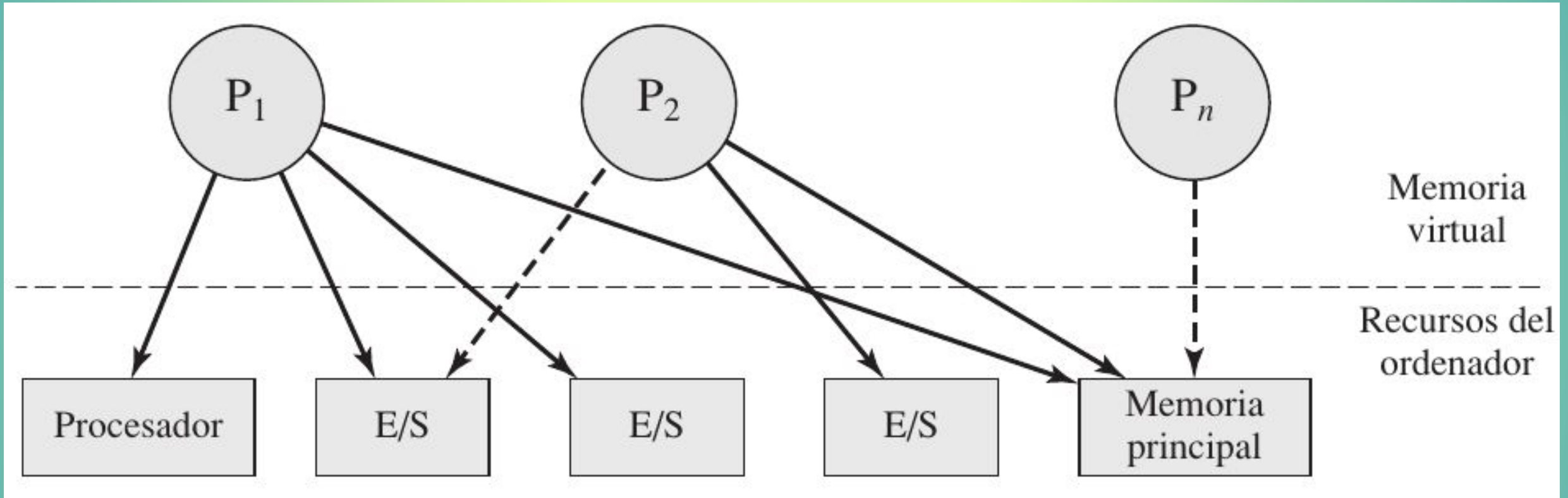


# Estructuras de datos y control de procesos

# Estructuras de datos del SO

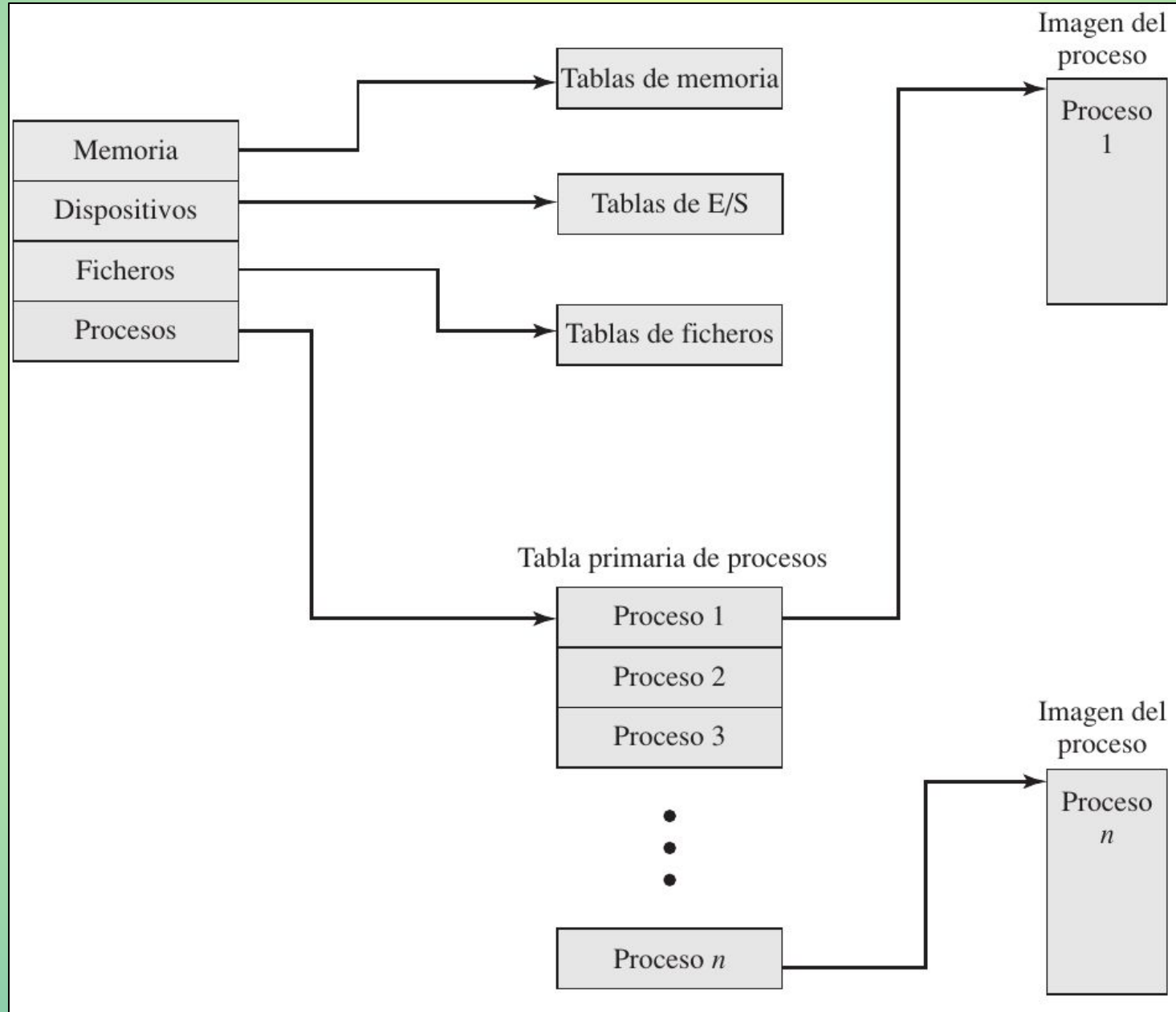
- El SO planifica y despacha procesos para ser ejecutados en el procesador
- Asigna recursos a los procesos
- Responde los pedidos de los programas de usuario

# Procesos y recursos



Instantánea de asignación de recursos

# Estructuras de control del SO. Tablas



# Tablas de memoria

- Asignación de memoria real a procesos
- Asignación de memoria secundaria a procesos
- Atributos de protección para acceso a regiones de memoria compartida
- Información necesaria para administrar memoria virtual

# Tablas de E/S

- Dispositivos de E/S disponibles o asignados
- Estado de operación de E/S
- Lugar de memoria principal usado como fuente o destino de la transferencia con el dispositivo de E/S



# Tablas de archivos

- Existencia de archivos
- Localidad en memoria secundaria
- Estado actual
- Atributos
- Algunas veces esta información la mantiene el sistema de manejo de archivos

# Tabla de procesos

Punteros a la información de cada proceso

- Imagen de proceso consiste en programa, datos, pila y atributos
- Atributos
  - bloque de control de proceso

# Bloque de control de proceso

- **Identificación de proceso**

- Identificador numérico único
  - puede ser un índice en la tabla de procesos primarios
- Identificador de usuario
  - quién es responsable del trabajo
- Identificador del proceso padre

- **Información de estado del procesador**

- Contenidos de los registros del procesador
  - Registros visibles al usuario
  - Registros de control y estado
  - Punteros de pila
- Palabra de estado de programa (PSW)
  - contiene información de estado
  - Ejemplo: el registro EFLAGS en máquinas Pentium

- **Información de control de procesos**

Información adicional necesaria para que el SO controle y coordine los diversos procesos activos

- información de planificación y estado
- estructuración de datos
- comunicación interprocesos
- privilegios del proceso
- manejo de memoria
- utilización y pertenencia de un recurso

# Control de procesos

- **Modos de ejecución**
  - Modo usuario
    - Menos privilegiado
    - Se ejecutan normalmente los programas de usuario en este modo
  - Modo sistema, modo control o modo kernel
    - más privilegiado
    - Se ejecuta el kernel del sistema operativo en este modo

# Funciones típicas de un núcleo de sistema operativo

## **Administración de procesos**

- Creación y terminación de procesos
- Planificación y despacho de procesos
- Conmutación de procesos
- Sincronización de procesos y soporte para comunicación entre procesos
- Administración de bloques de control de procesos

## **Manejo de memoria**

- Asignación del espacio de direcciones a los procesos
- Swapping
- Manejo de segmentos y páginas

## **Manejo de E/S**

- Manejo de buffer
- Asignación a procesos de canales de E/S y dispositivos

## **Funciones soportadas**

- Manejo de interrupciones
- Contabilidad
- Monitoreo



# Creación de procesos

- Asignar un identificador único de proceso
- Asignar espacio al proceso
- Inicializar el bloque de control de proceso
- Establecer los vínculos apropiados
  - Ej: agregar nuevo proceso a la lista usada para la cola de planificación
- Otros
  - mantener un archivo de estadísticas



# Conmutación de procesos

- Cuestiones de diseño
  - ¿Qué sucesos provocan un cambio de proceso?
  - ¿hay que distinguir entre cambio de modo y cambio de proceso?
  - ¿qué debe hacer el so con las diferentes estructuras de datos bajo su control para llevar a cabo un cambio de proceso?

# Conmutación de procesos

## Sucesos para que el so tome control

- **Interrupciones** (Externas a la ejecución de la instrucción actual)
  - Reloj
    - proceso ha usado el máximo tiempo de ejecución permitido por vez
  - E/S
  - Fallo de memoria
    - la dirección de memoria está en memoria virtual, debe ser traída a memoria real
- **Trampa** (Asociada con la ejecución de la instrucción actual)
  - ocurrencia de un error
  - puede ser la causa de que un proceso se mueva al estado de terminado
- **Llamada a supervisor** (pedido explícito)
  - por ejemplo abrir un archivo

# Conmutación de procesos

## Cambio de modo

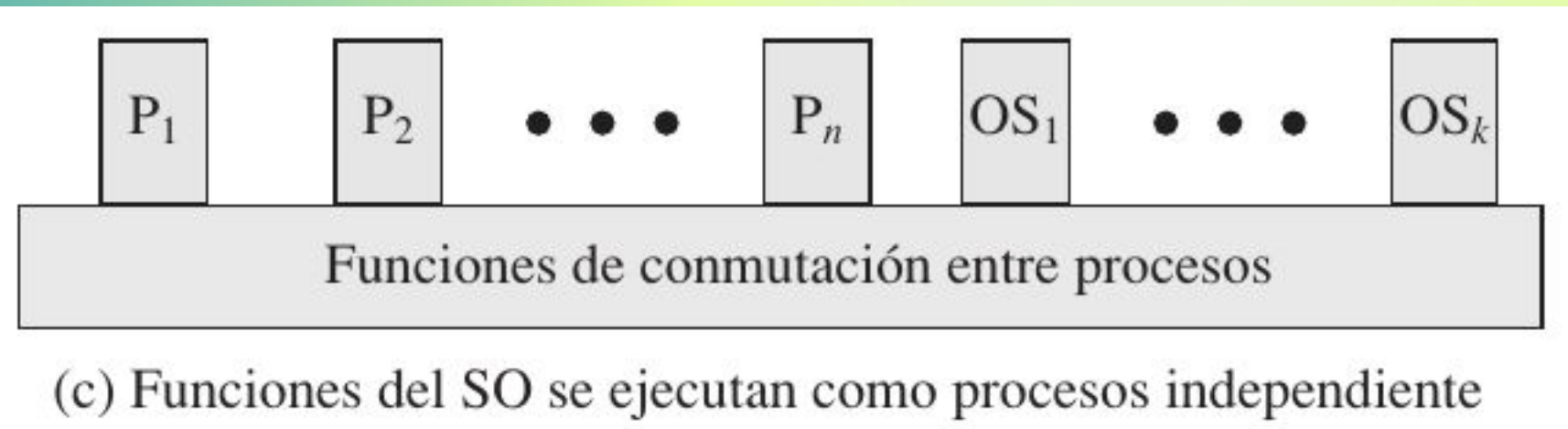
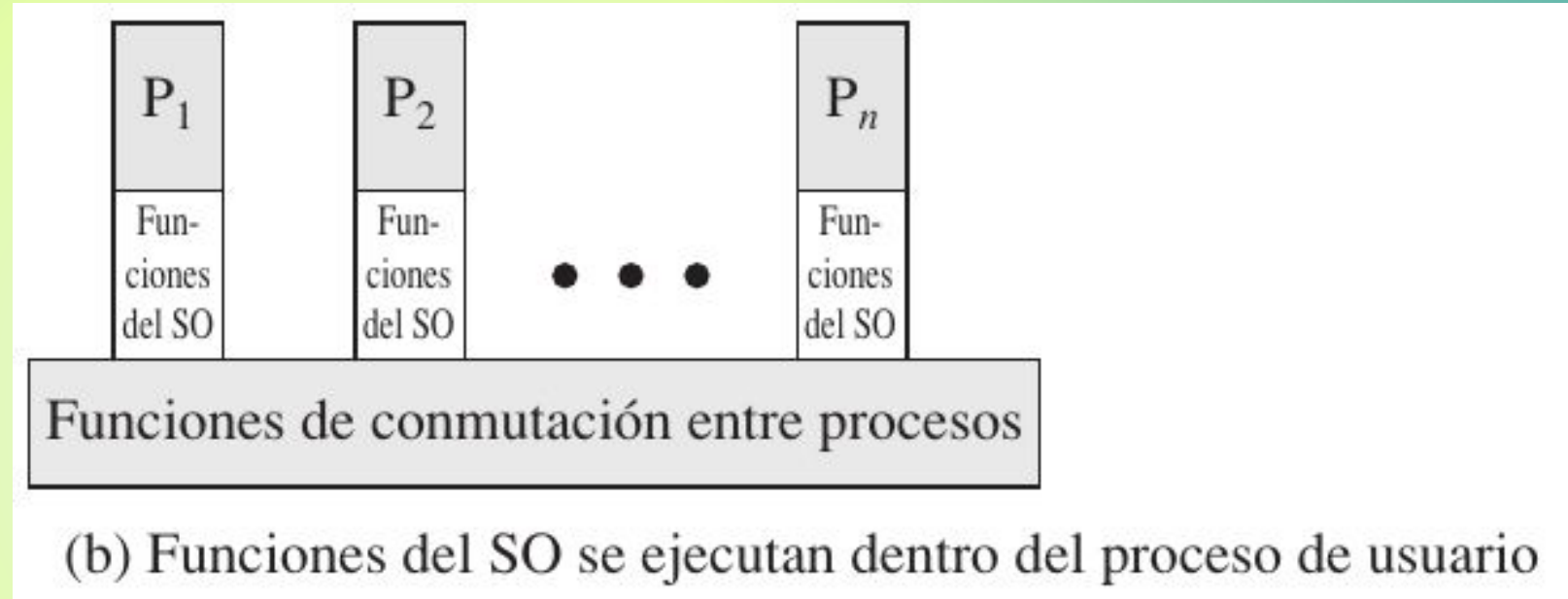
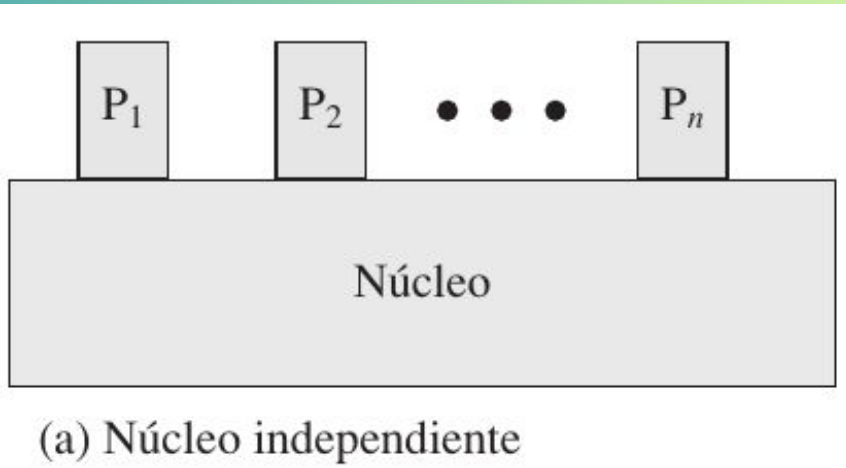
- Cuando hay una interrupción pendiente el procesador
  - Salva contexto del programa que está ejecutando
    - La parte del BCP de información de estado del procesador
  - Cambia el contador de programa al que corresponde a la rutina de interrupciones
  - Cambia de modo de usuario a modo de núcleo
  - Ejecuta rutina de interrupciones y al terminar esta se puede
    - Reanudar ejecución del mismo proceso o
    - Cambiar de proceso

# Conmutación de procesos

## Cambio completo de proceso

- Guardar contexto de procesador incluyendo contador de programa y otros registros
- Actualizar el bloque de control de proceso con el nuevo estado y cualquier información contable
- Mover bloque de control de proceso a la cola apropiada listo, bloqueado
- Seleccionar otro proceso a ejecutar
- Actualizar el bloque de control de proceso del proceso seleccionado
- Actualizar estructuras de datos de manejo de memoria
- Restaurar contexto del proceso seleccionado

# Ejecución del sistema operativo

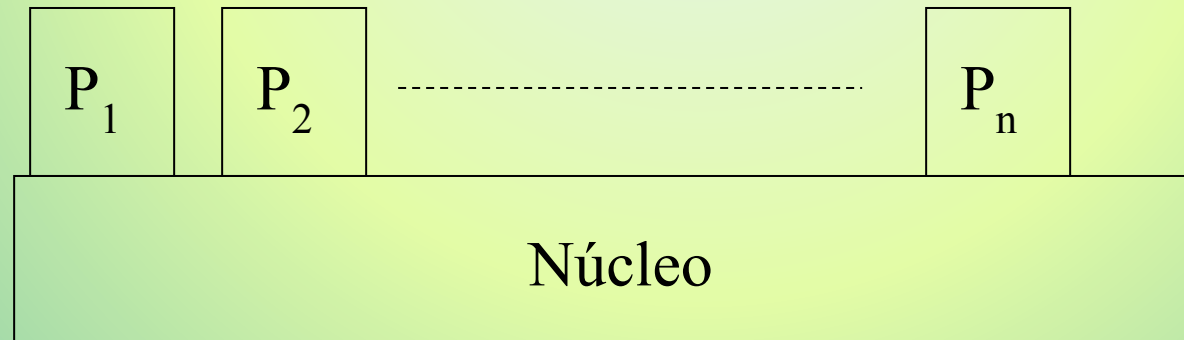




# Ejecución del sistema operativo

- **Kernel sin proceso**

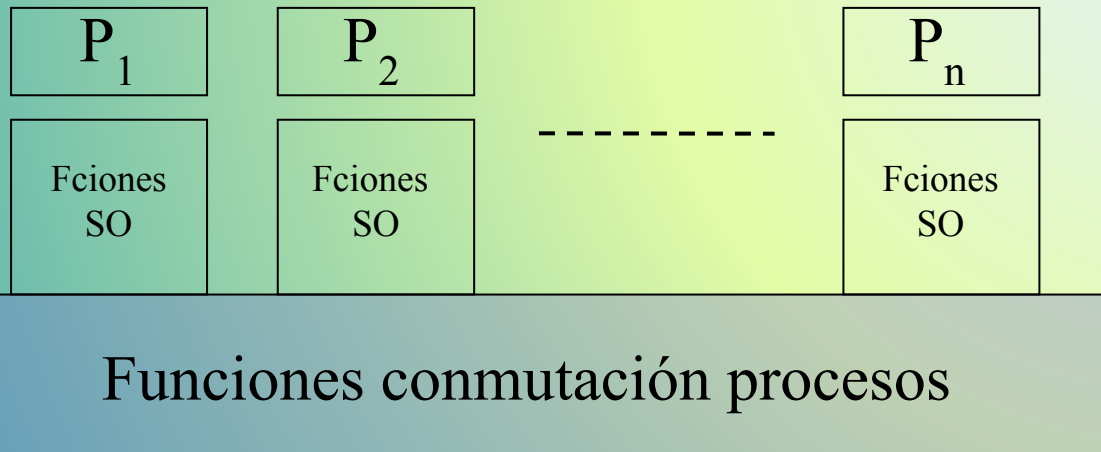
- ejecutar kernel fuera de cualquier proceso
- el código del sistema operativo se ejecuta como una entidad separada que opera en modo privilegiado
- proceso: programa de usuarios



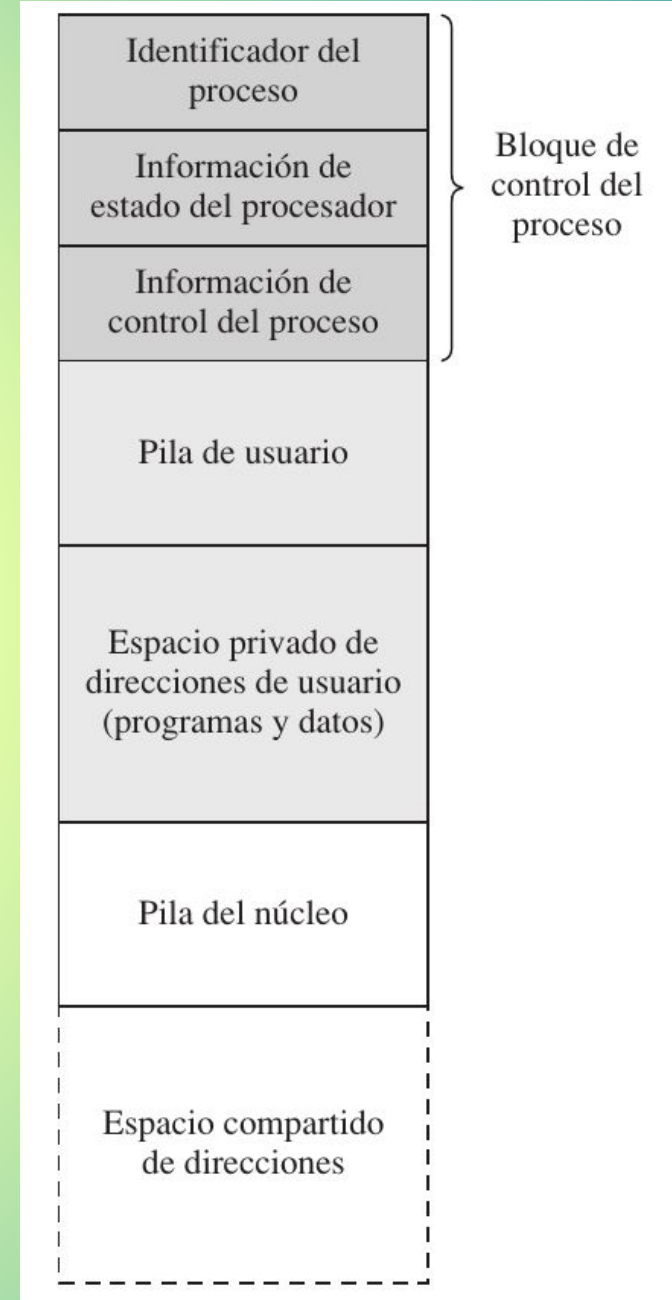


# Ejecución del sistema operativo

- **Ejecución dentro de procesos de usuarios**
  - software de SO dentro del contexto de los procesos de usuario
  - proceso se ejecuta en modo privilegiado cuando se ejecuta código de SO

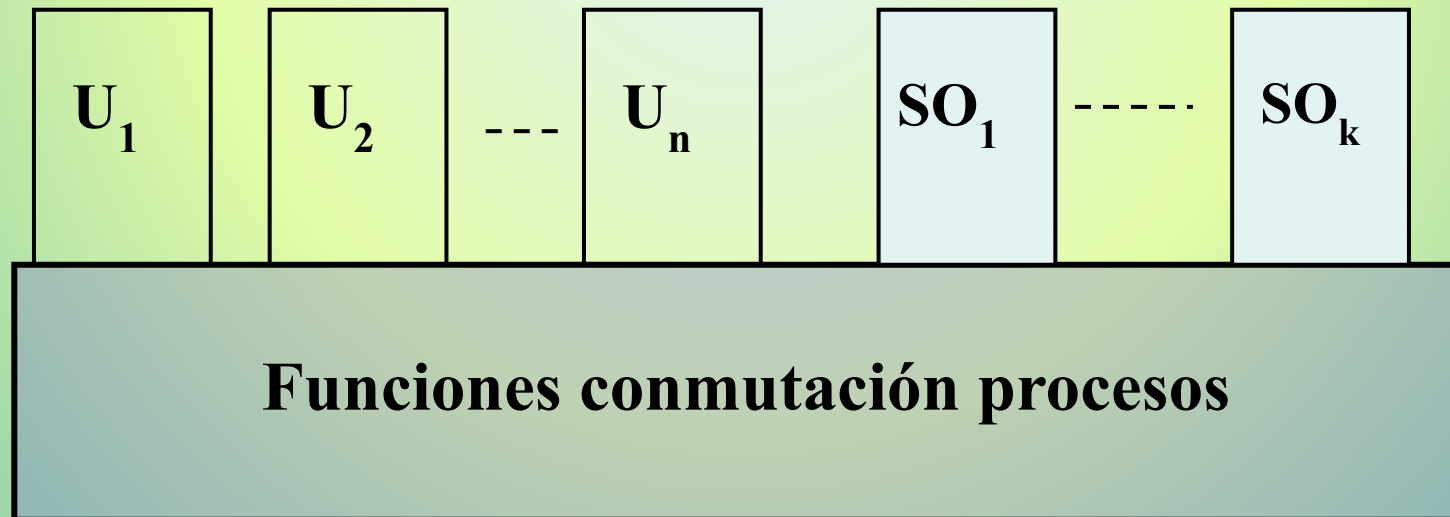


→  
Imagen de proceso:  
el SO ejecuta dentro  
del espacio de usuario.

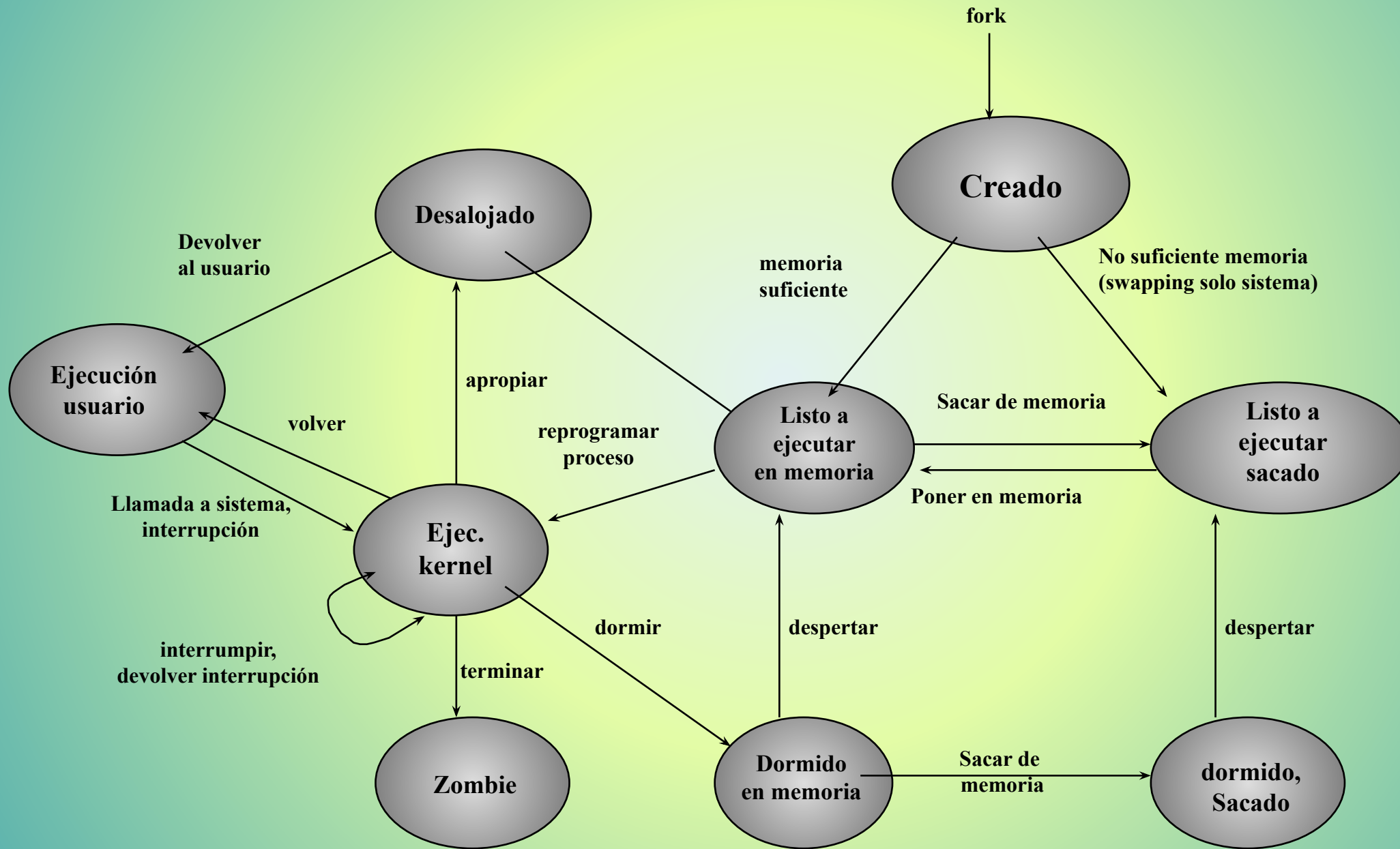


# Ejecución del sistema operativo

- **Sistema Operativo basado en procesos**
  - funciones principales del kernel son procesos separados
  - el sistema operativo invoca a un proceso



# Diagrama de transición de estados en UNIX



Hilos

# Procesos

- Pertenencia de recursos - se le asigna al proceso un espacio de direcciones virtual para contener la imagen del proceso, también a veces la memoria principal y control de otros recursos
- Despacho - proceso es un camino de ejecución a través de uno o más programas
  - se puede intercalar la ejecución con otros procesos
- El sistema operativo trata estas dos características en forma independiente

# Procesos

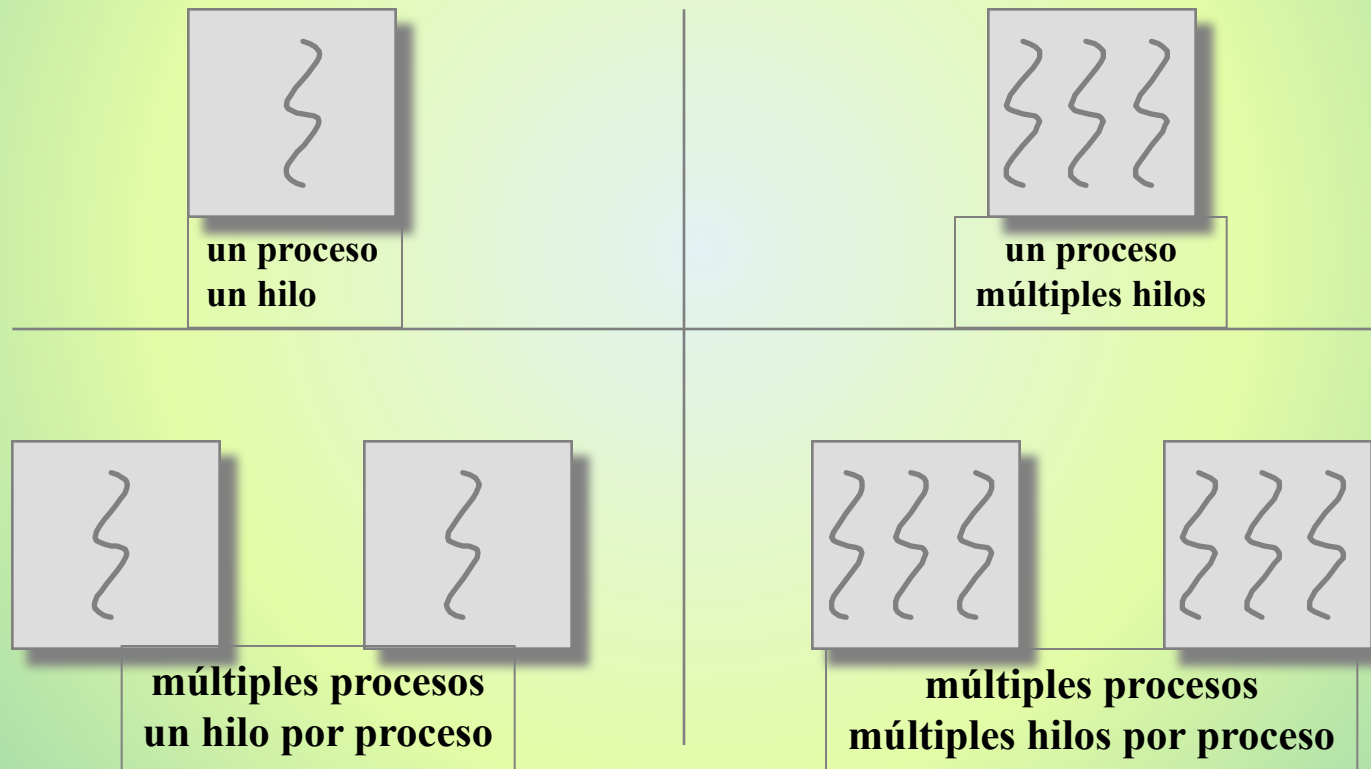
- Hilo se refiere al despacho
- Proceso o tarea se refiere a la pertenencia de los recursos



# Multihilos

- El sistema operativo soporta hilos múltiples de ejecución dentro de un solo proceso
- MS-DOS soporta un único hilo
- UNIX soporta múltiples procesos de usuarios pero sólo un hilo por proceso
- Windows, solaris y otros soportan hilos múltiples

# Hilos y procesos



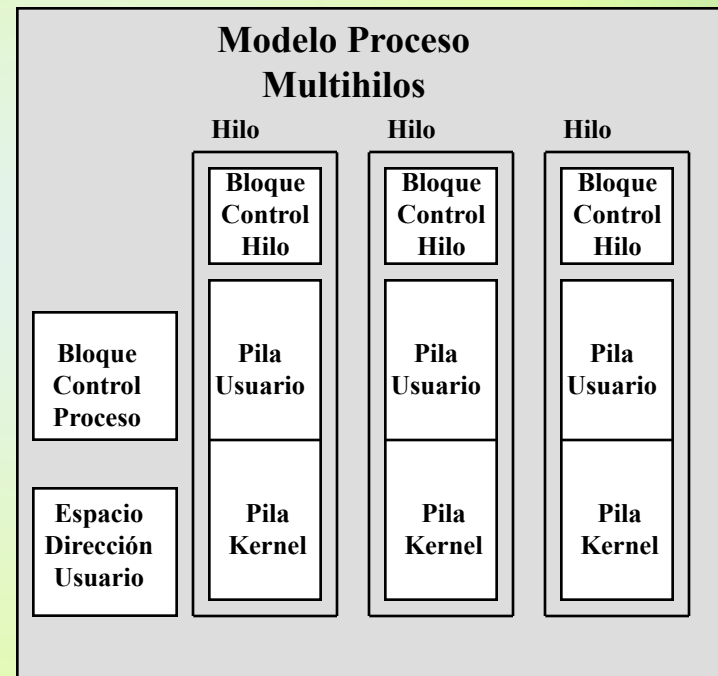
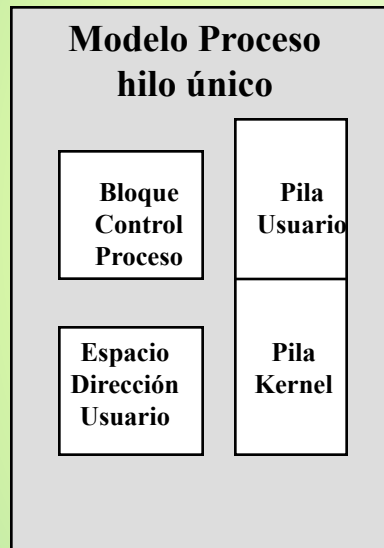
# Procesos

- Tienen un espacio de direcciones virtual que contiene la imagen del proceso
- Acceso protegido a los procesadores, otros procesos, archivos y recursos de E/S

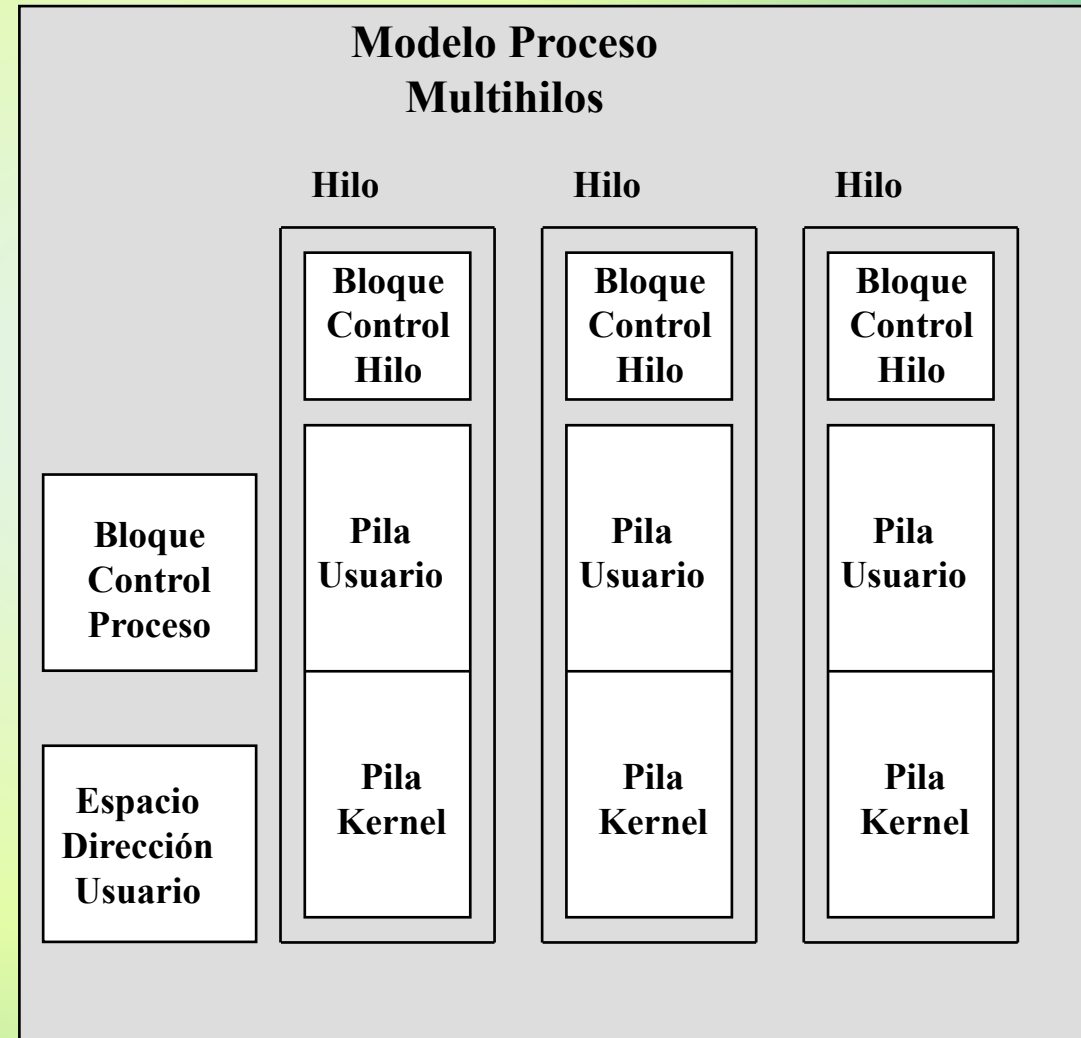
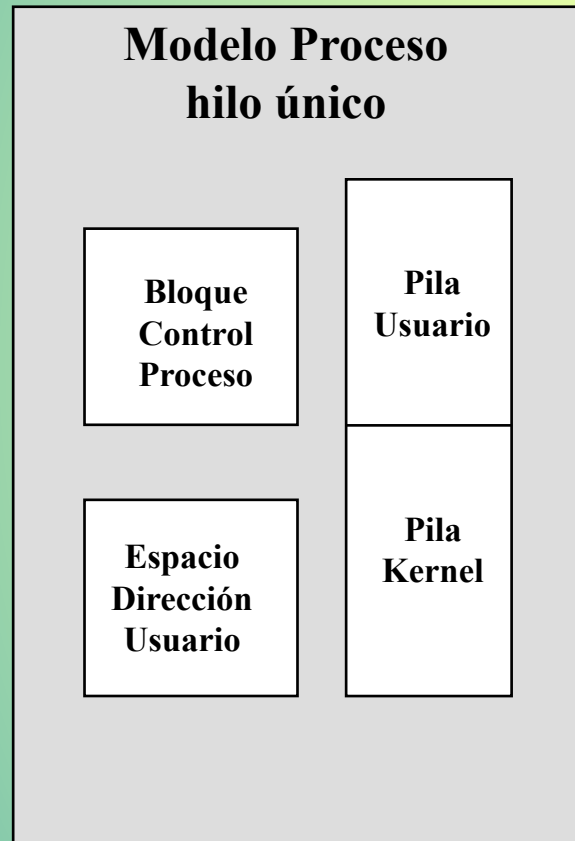
# Hilos

- Tiene un estado de ejecución (ejecución, listo, etc)
- Guarda contexto de hilo cuando no se está ejecutando
- Tiene una pila de ejecución
- Tiene algún almacenamiento estático por hilo para variables locales
- Tiene acceso a la memoria y recursos de su proceso
  - todos los hilos de un proceso comparten esto

# Modelos de procesos de uno y múltiples hilos



# Modelos de procesos de uno y múltiples hilos





# Beneficios de los hilos

- Lleva menos tiempo crear un hilo nuevo que un proceso
- Menor tiempo para terminar un hilo que un proceso
- Menor tiempo cambiar entre dos hilos dentro del mismo proceso
- Ya que los hilos dentro del mismo proceso comparten memoria y archivos, se pueden comunicar entre sí sin invocar al kernel

# Hilos

- Suspender un proceso implica suspender todos los hilos del proceso ya que todos los hilos comparten el mismo espacio de direcciones
- Al terminar un proceso se terminan todos los hilos dentro del proceso

# Hilos. Funcionalidad

- Estados de los hilos
  - Ejecución, listo, bloqueado
- Operaciones básicas
  - Crear (spawn)
  - Bloquear
  - Desbloquear (despertar)
  - Terminar
- Sincronización de hilos

# Hilos de nivel usuario

- Todo el manejo de hilos se hace a través de la aplicación
- El kernel no sabe de la existencia de hilos
- Intercambio de hilos no requiere modo privilegiado de kernel
- La planificación es específica a la aplicación

# Hilos de nivel núcleo

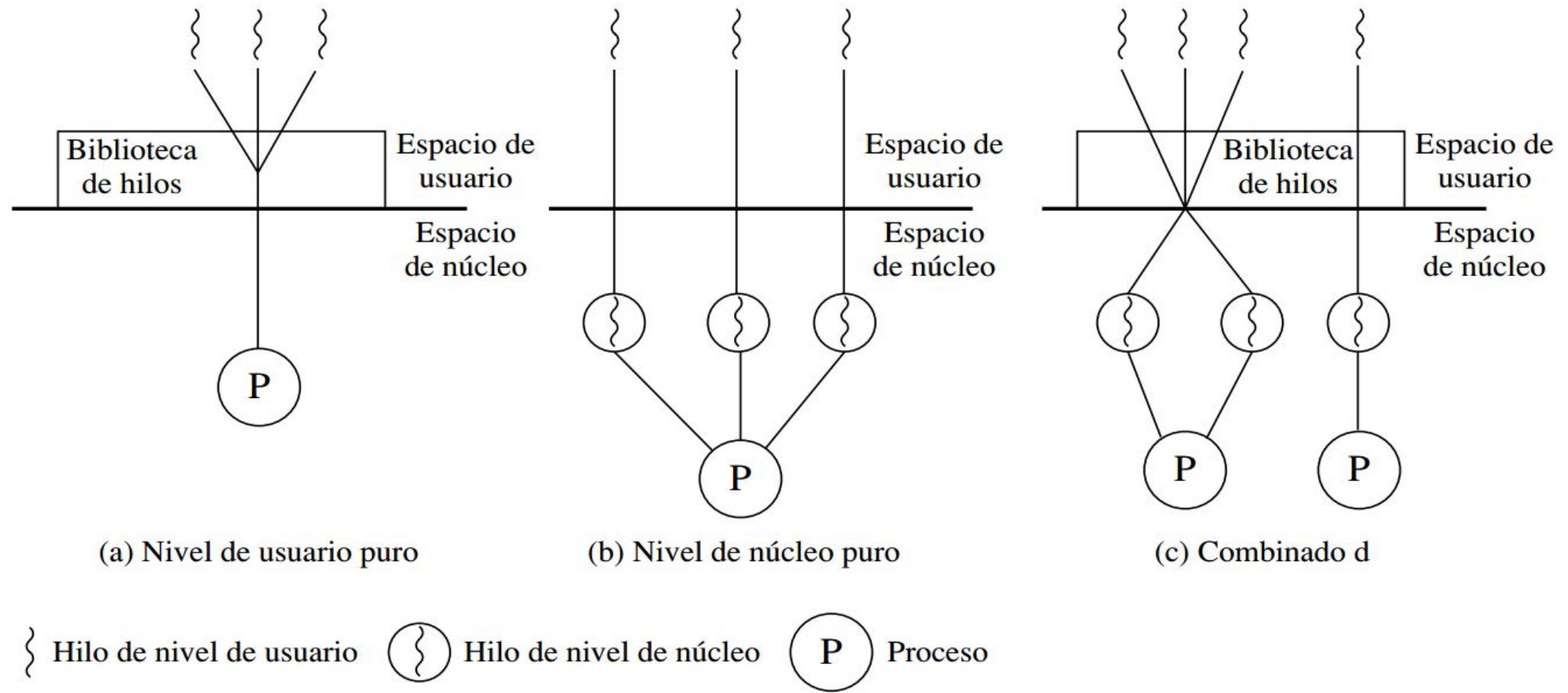
- Windows 2000, Linux y OS/2 son ejemplos
- El núcleo mantiene la información de contexto para el proceso y los hilos
- Cambiar entre hilos requiere kernel

# Propuestas combinadas para hilos

- Solaris es el ejemplo
- Creación de hilos hecha en el espacio de usuario
- El grueso de la planificación y sincronización de hilos se hace en el espacio de usuario



# Hilos nivel usuario y nivel kernel

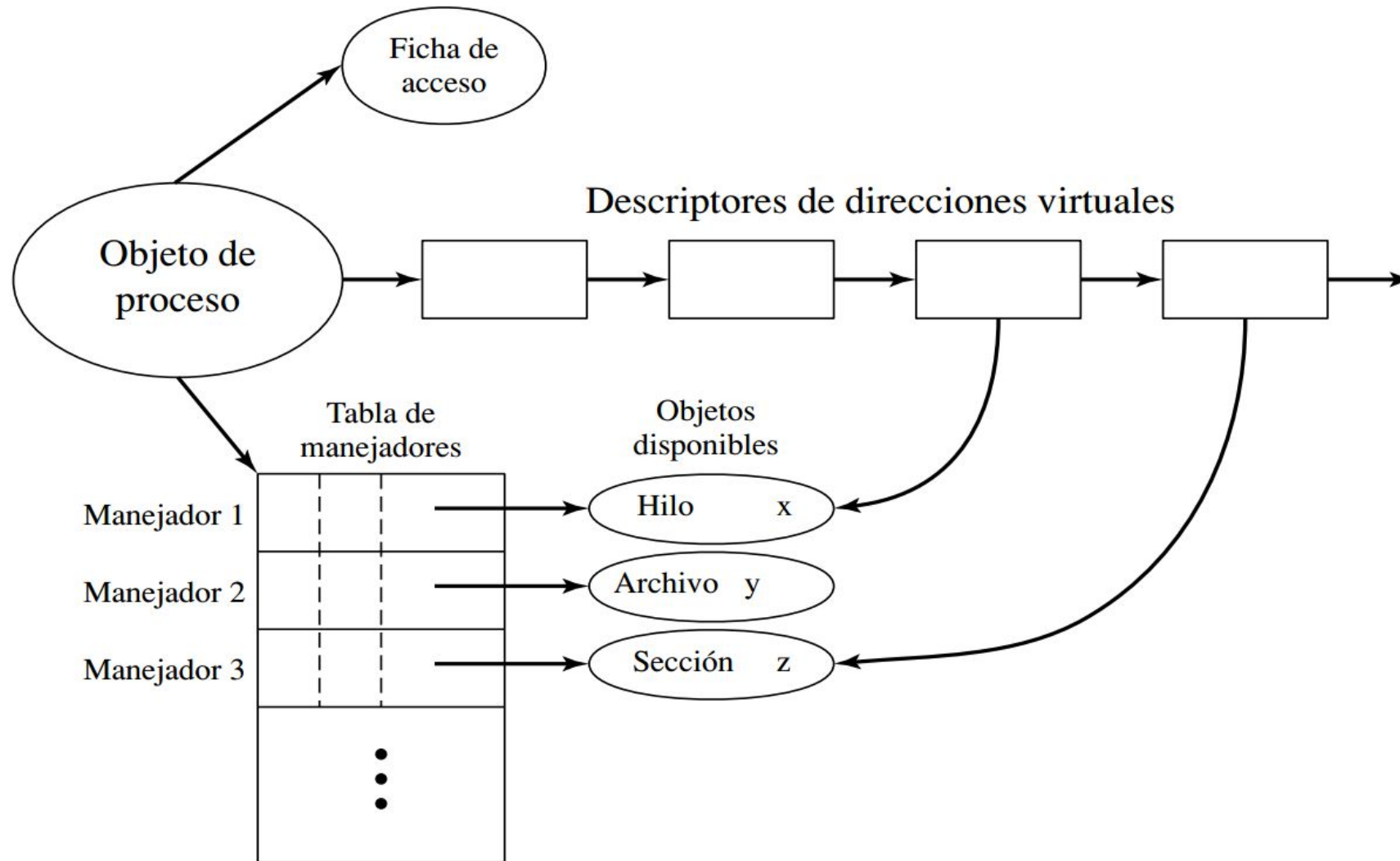


**Figura 4.6.** Hilos de nivel de usuario y de nivel de núcleo.

# Procesos Windows

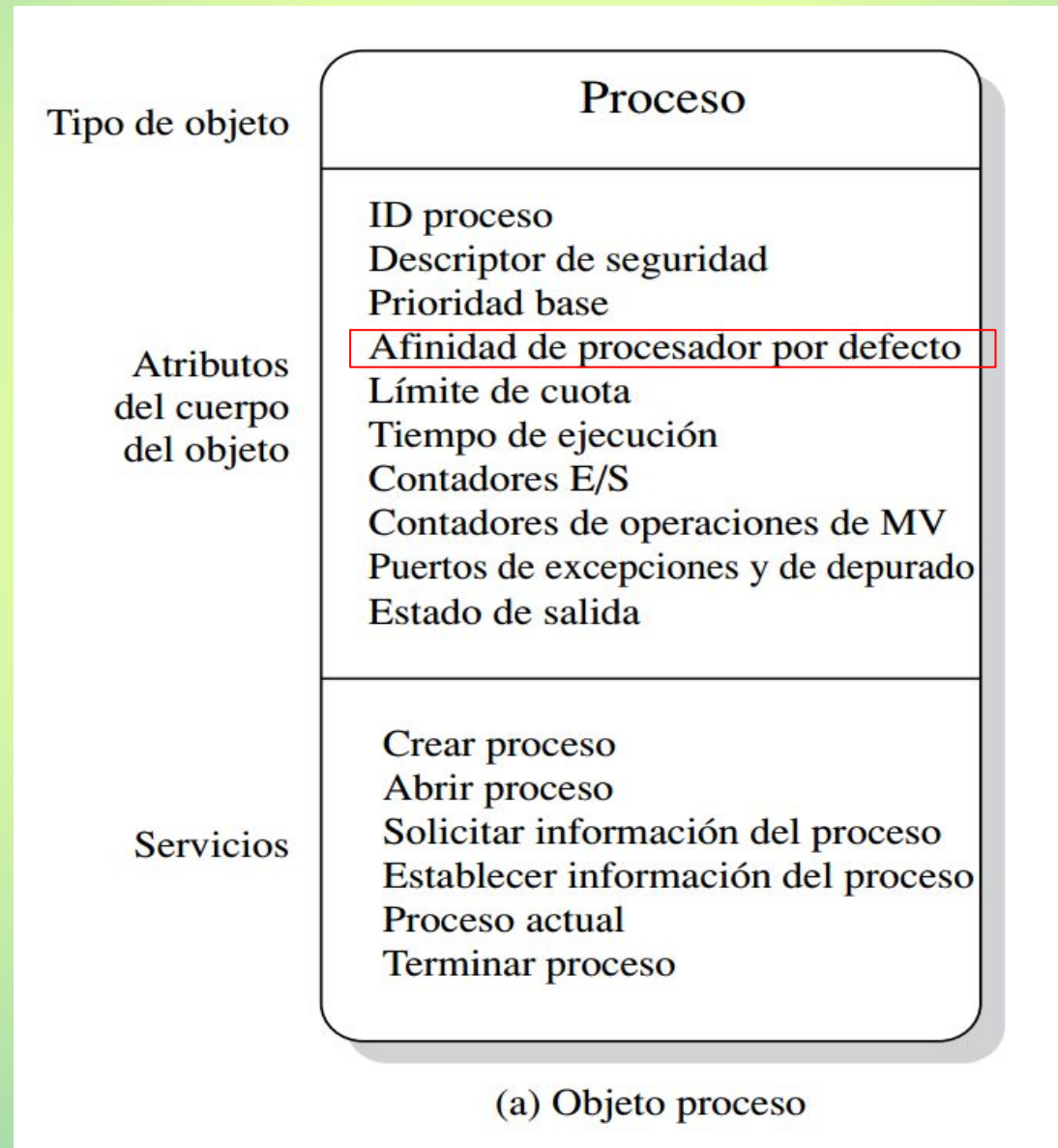
- Implementados como objetos
- Un proceso ejecutable puede contener uno o más hilos
- Ambos objetos procesos e hilos tienen capacidades de sincronización incorporadas

# Procesos Windows

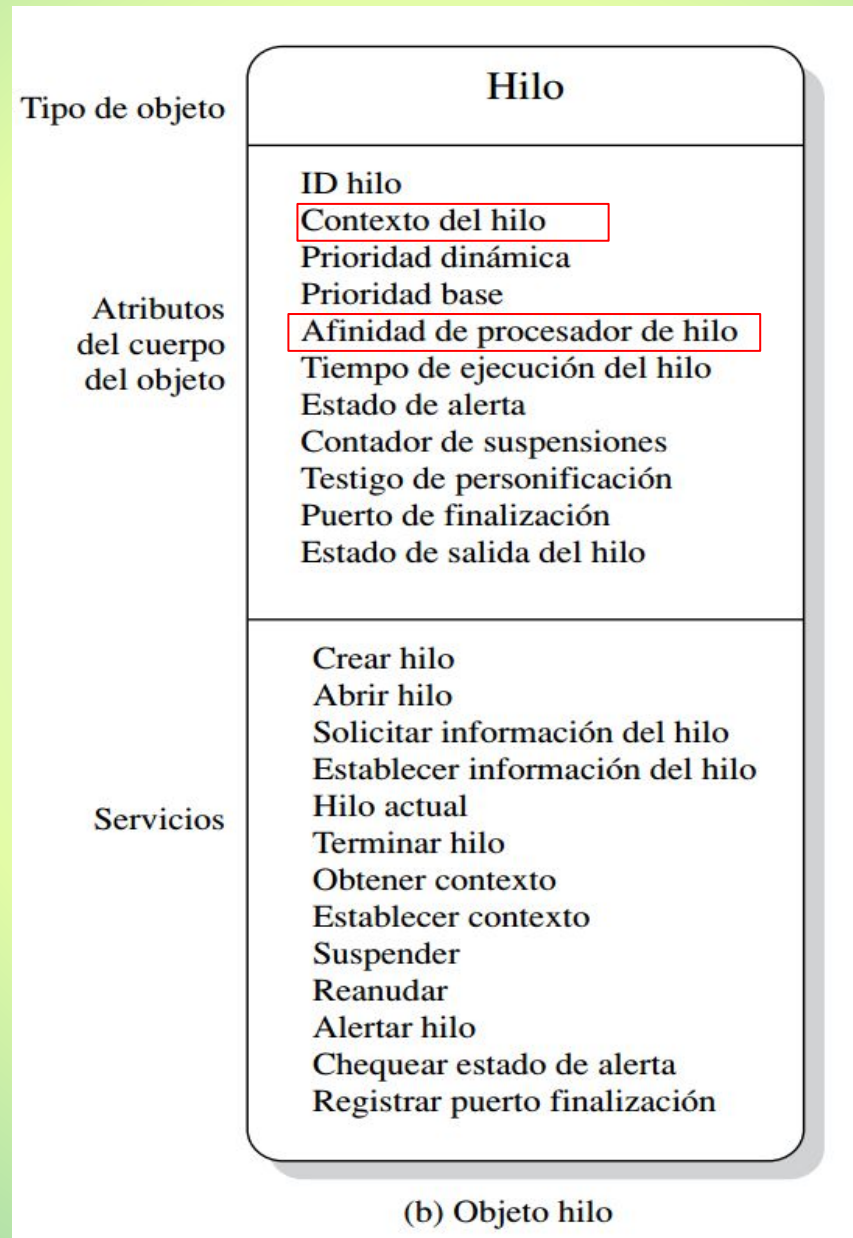


**Figura 4.12.** Un proceso Windows y sus recursos.

# Objeto proceso en Windows

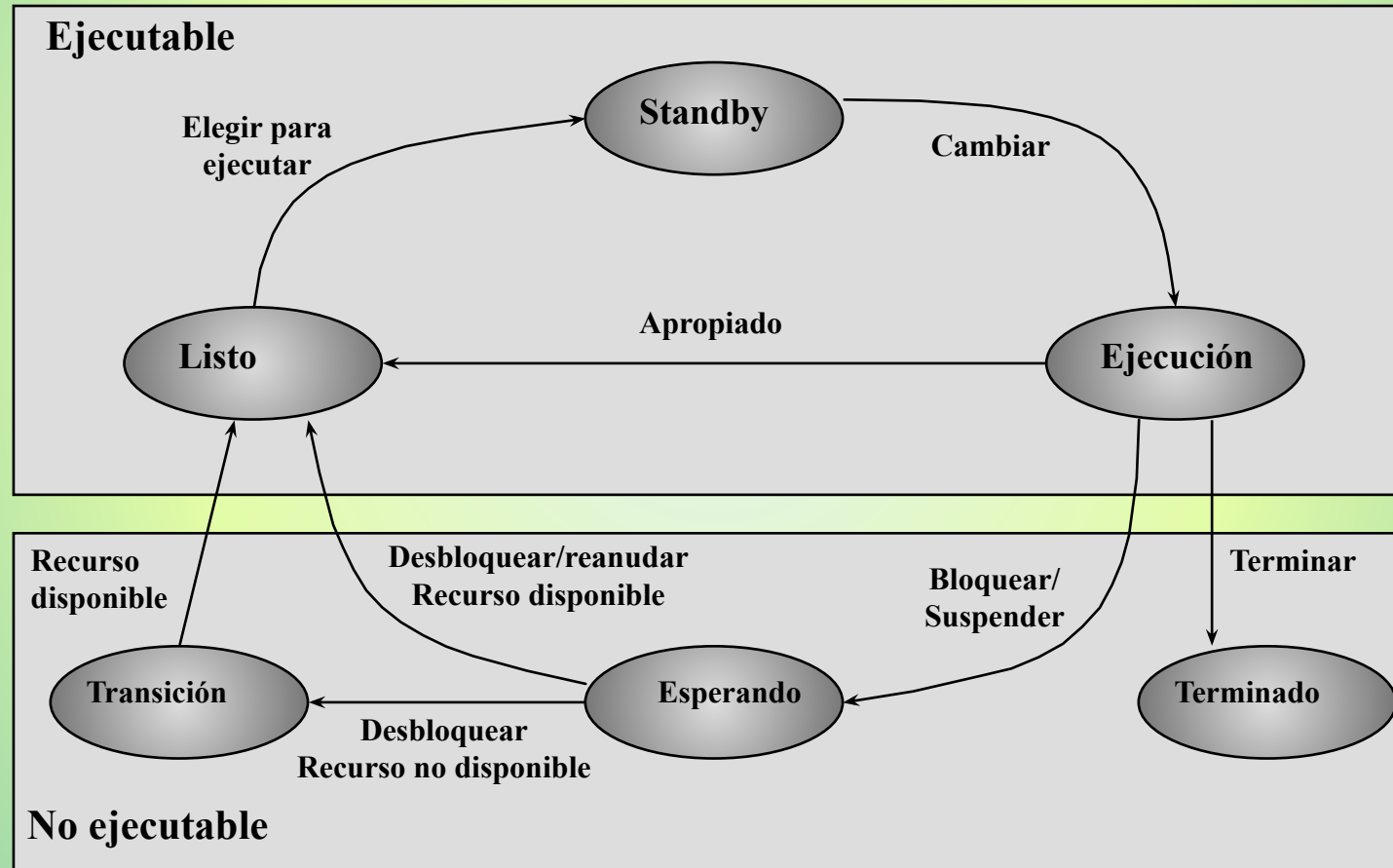


# Objeto hilo en Windows





# Estado de hilos Windows

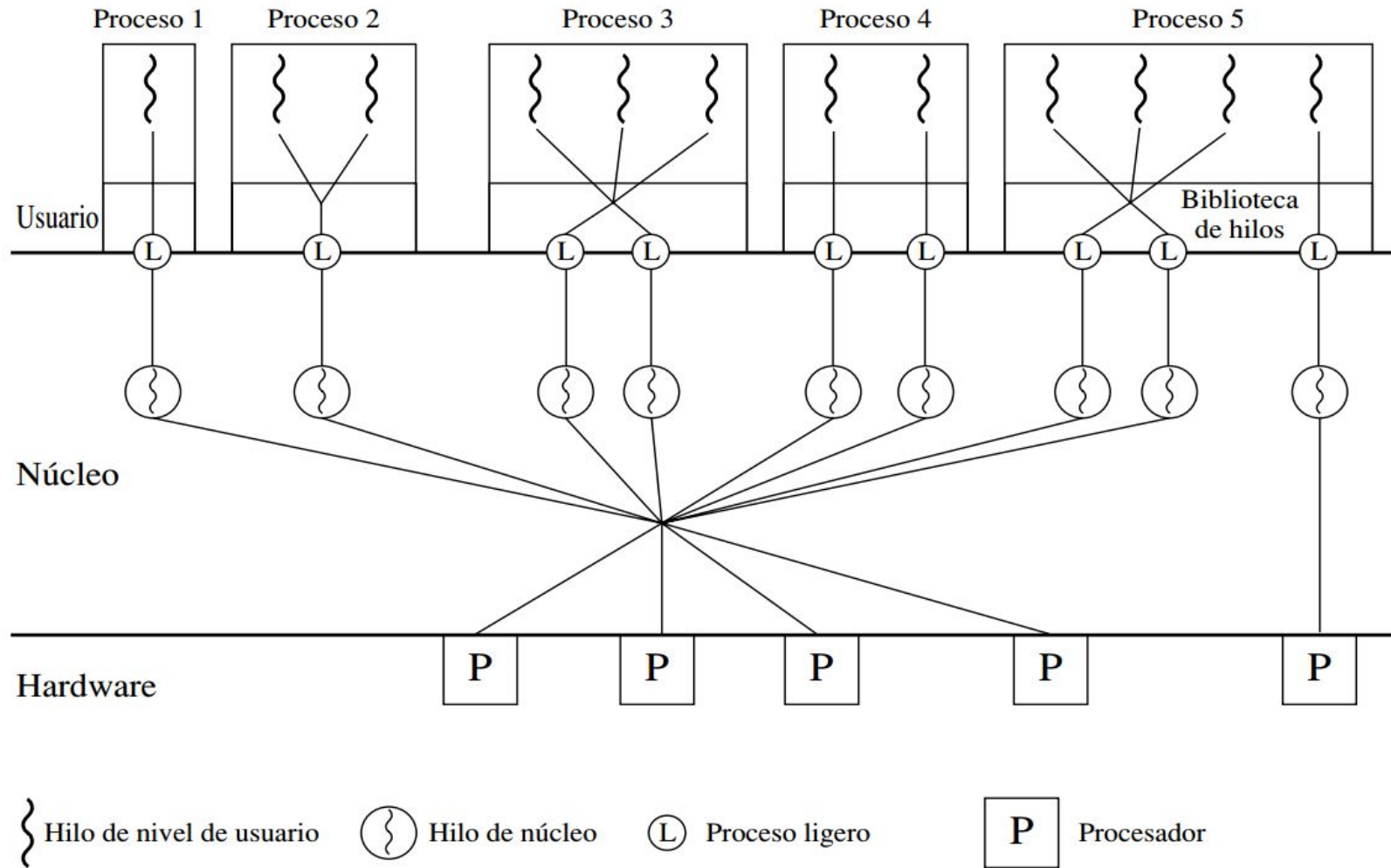




# Solaris

- **Proceso** incluye el espacio de direcciones del usuario, pila y bloque de control de proceso
- **Hilos multiusuario**
- **Procesos livianos**
- **Hilos de núcleo**

# Solaris



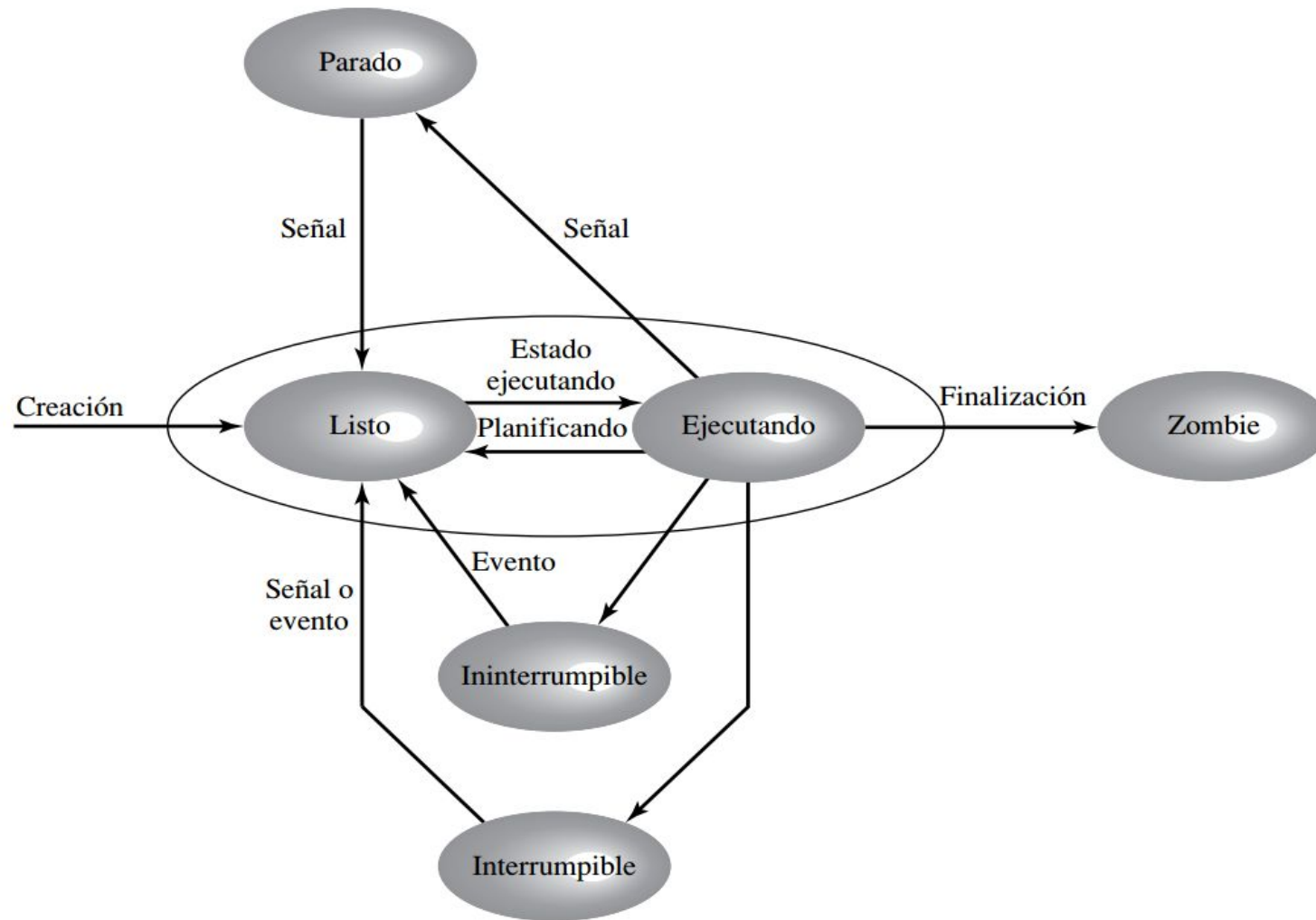
**Figura 4.15.** Ejemplo de arquitectura multihilo de Solaris.

# Procesos en Linux

- Estado
- Información planificación
- Identificadores
- Comunicación interprocesos
- Links
- Tiempos y cronómetros
- Sistema de archivos
- Memoria virtual
- Contexto específico al procesador

# Linux

## Modelo de procesos/hilos



**Figura 4.18.** Modelo de procesos e hilos en Linux.