Tema 3

Descripción y Control de Procesos

Stallings: 3.1 – 3.4

Contenido

- Introducción: qué es un proceso.
- Estados de un proceso: modelos de colas.
- Creación y terminación de procesos.
- Descripción de procesos: estructuras de control.
- Control de procesos.
- Modos de ejecución.

Principales requisitos de los sistemas operativos

- Intercalar la ejecución de múltiples procesos para maximizar la utilización del procesador ofreciendo a la vez un tiempo de respuesta razonable.
- Asignar los recursos a los procesos.
- Dar soporte a la comunicación entre procesos y la creación de procesos por parte del usuario.

Proceso

- También se llama tarea.
- Ejecución de un programa individual.
- Traza del proceso:
  - Listado de la secuencia de instrucciones que se ejecutan para dicho proceso.
- Dintinguir entre Proceso (concepto dinámico) y Programa (concepto estático).

Proceso = Código (texto)
+ Actividad (contador, registros, ficheros, E/S)
+ Pila (datos temporales, parámetros,...)
+ Datos (globales)

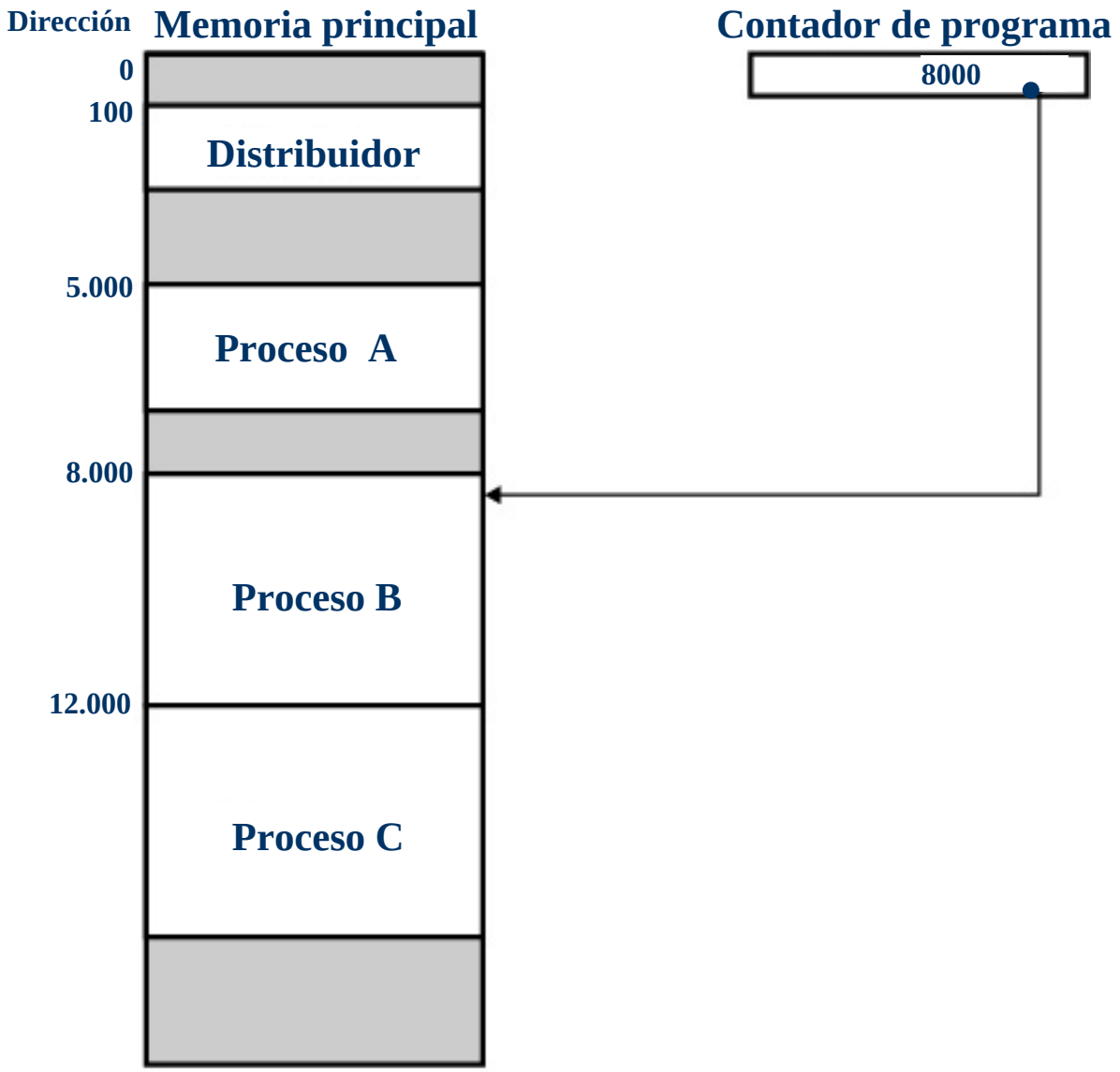
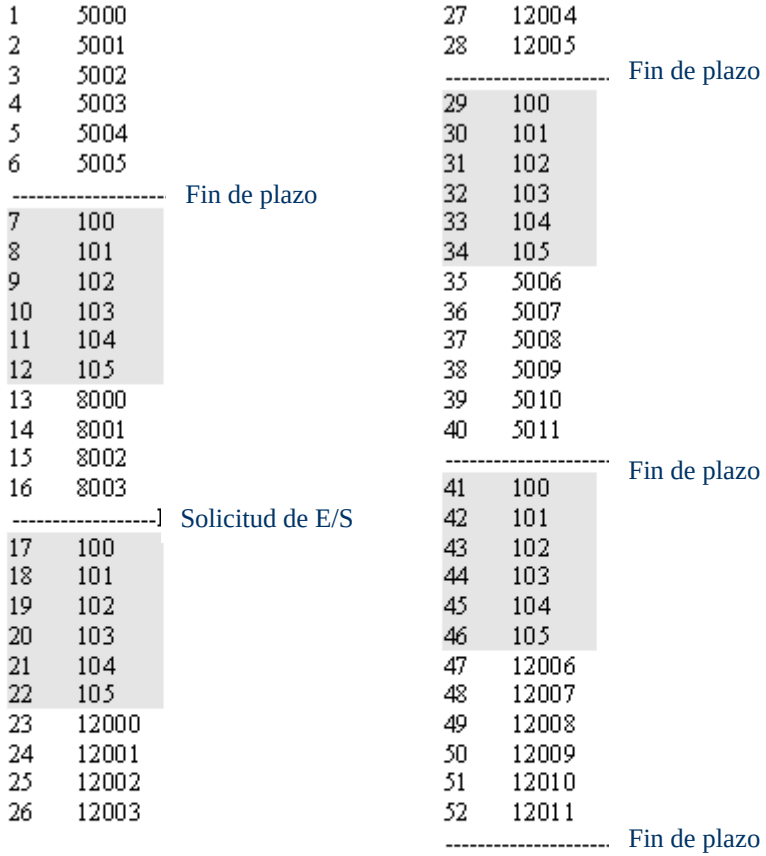


Figura 3.1. Instantánea de un ejemplo de ejecución (Figura 3.3) en el ciclo de instrucción 13.

(a) Traza del proceso A	(b) Traza del proceso B	(c) Traza del proceso C
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

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

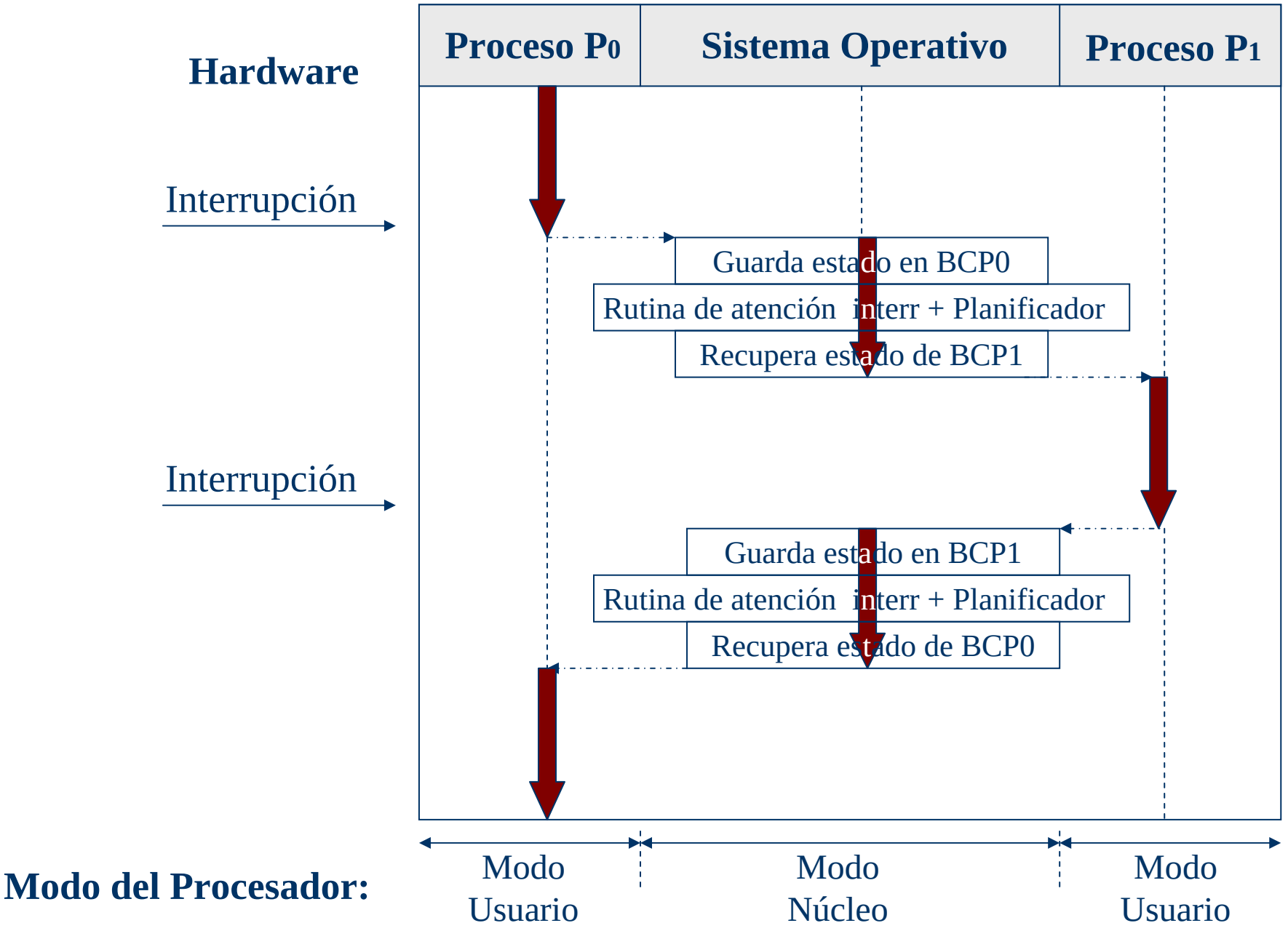
Figura 3.2. Trazas de los procesos de la Figura 3.1.



100 = Dirección de comienzo del programa distribuidor
Las áreas sombreadas indican ejecución del proceso distribuidor;
la primera y tercera columna cuentan los ciclos de instrucción;
la segunda y cuarta columna muestran la dirección de la
instrucción a ejecutar.

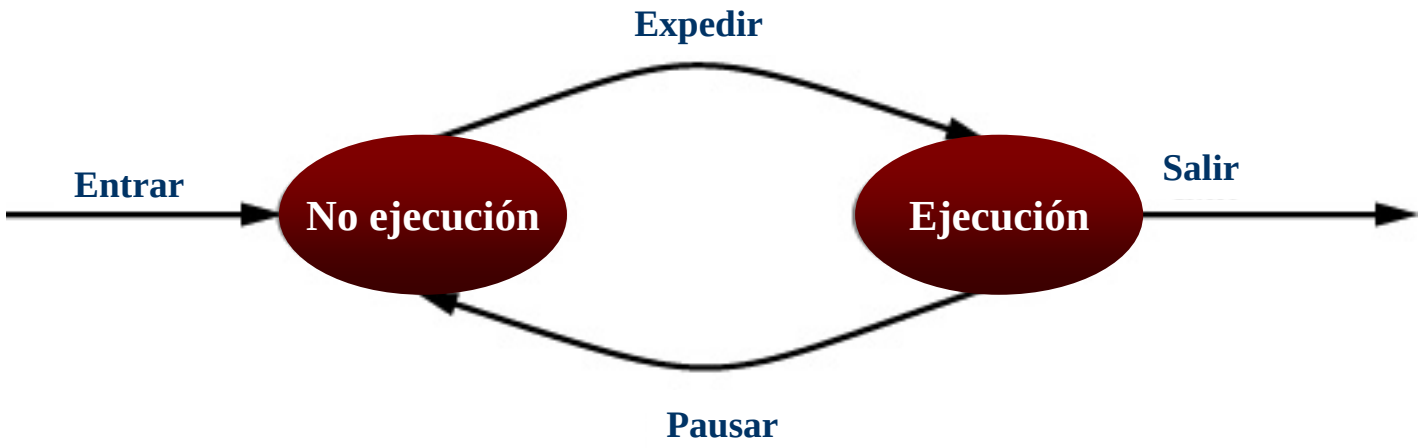
Figura 3.3. Traza combinada de los procesos de la Figura 3.1.

- Procedimiento empleado en **Multiprogramación** y **Tiempo Compartido**



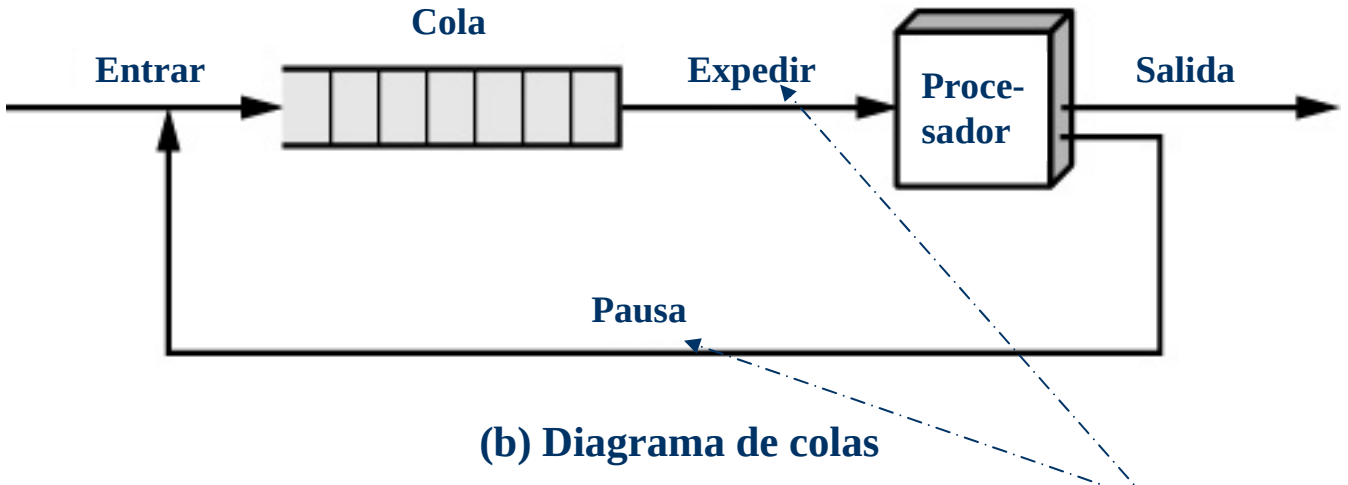
Modelo de proceso con dos estados

- Un proceso puede estar en uno de dos estados:
  - Ejecución.
  - No Ejecución.



(a) Diagrama de transición de estados

Proceso en estado de No Ejecución en una cola



(b) Diagrama de colas

Planificador a corto plazo (*Dispatcher*)

Procesos

- No Ejecución:
  - Listos para ejecutarse.
- Bloqueado:
  - Esperan a que termine una operación de E/S.
- El distribuidor podría no seleccionar exactamente el proceso que está en el extremo más antiguo de la cola porque podría estar bloqueado.

Un modelo de cinco estados

- Ejecución.
- Listo.
- Bloqueado.
- Nuevo.
- Terminado.

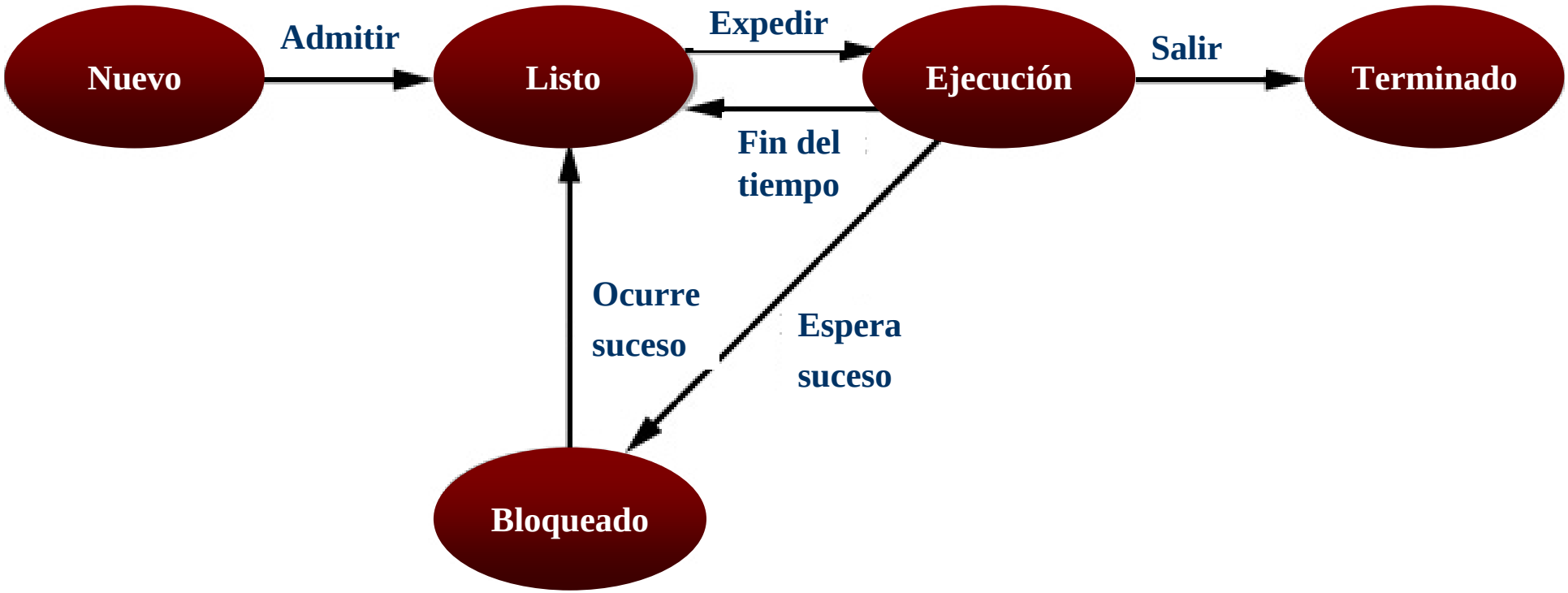


Figura 3.5. Modelo de procesos de cinco estados.

Diagramas de Gantt:

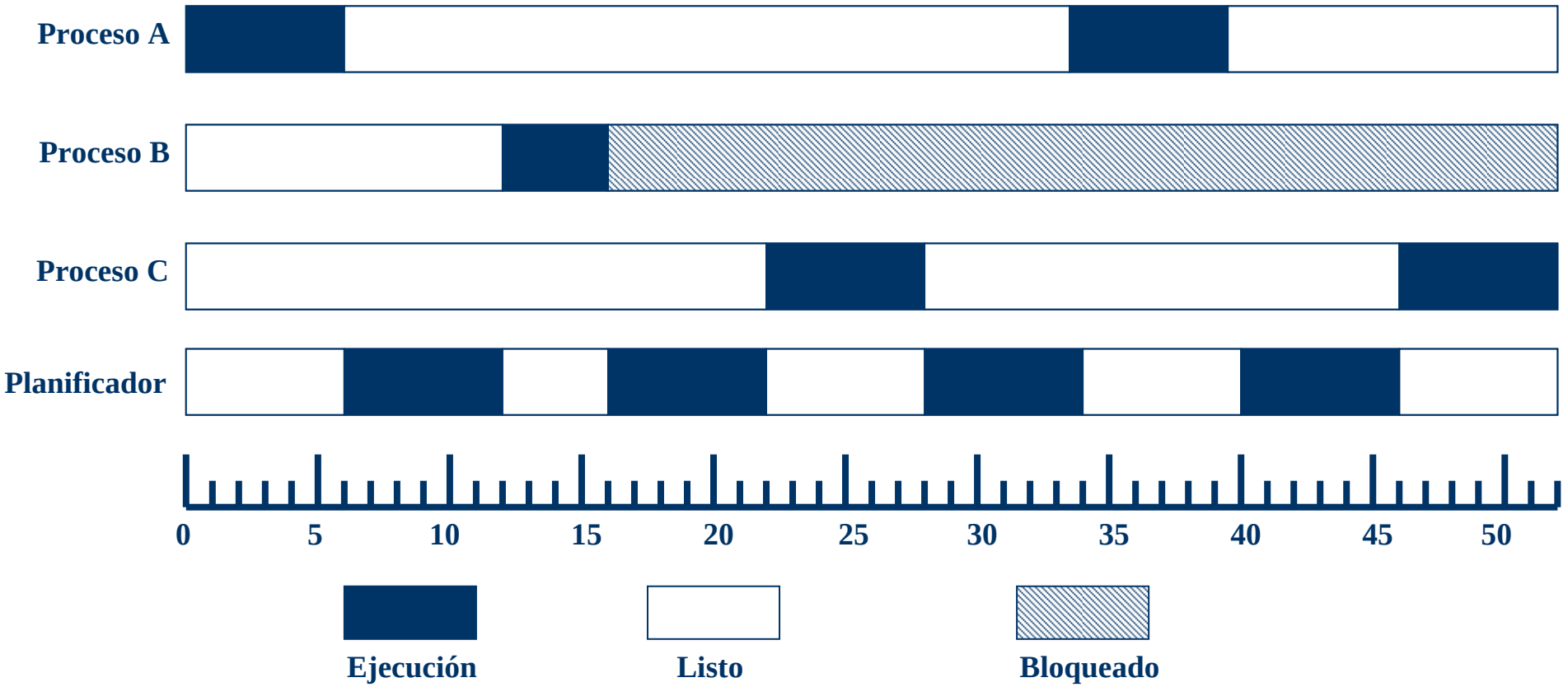
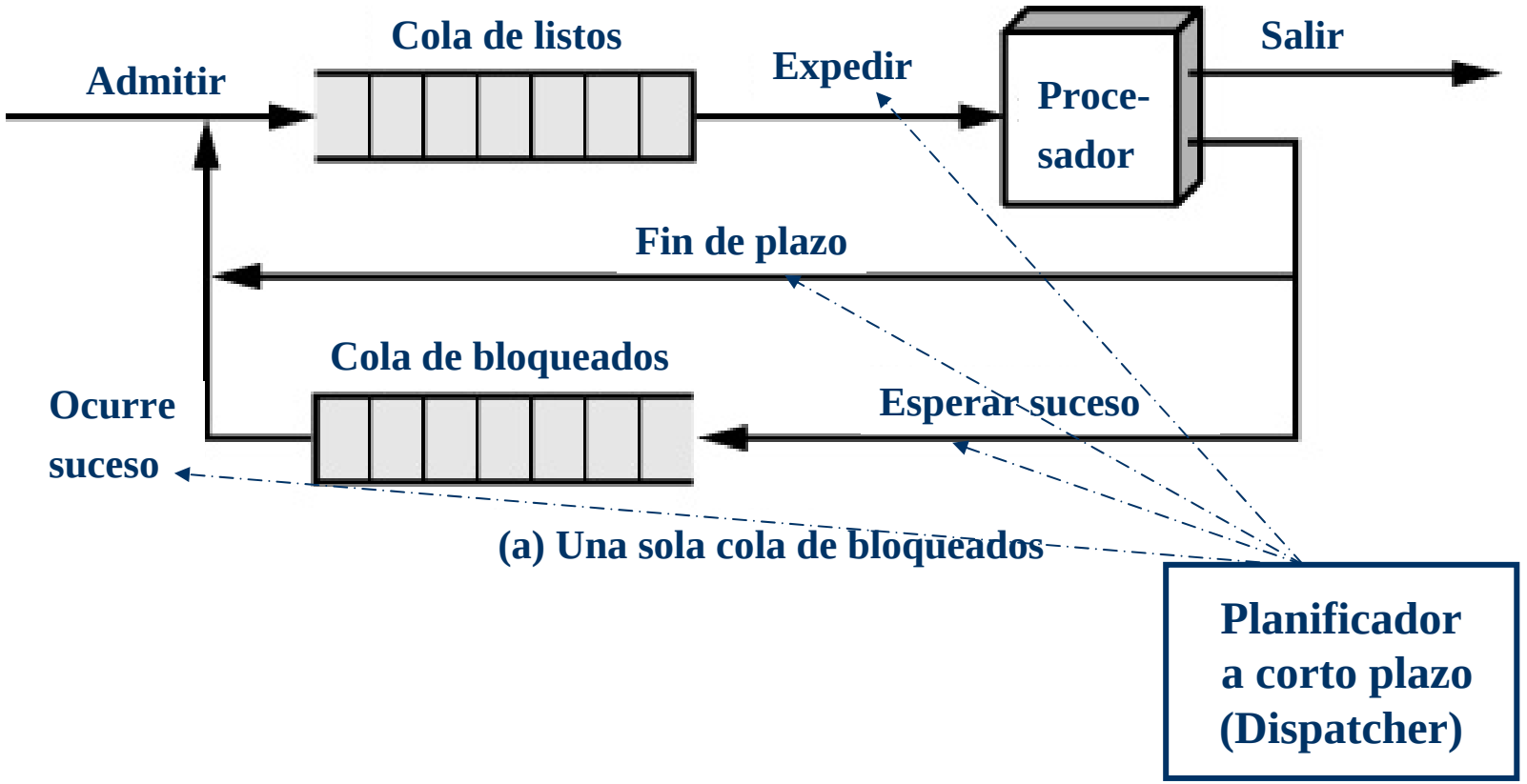
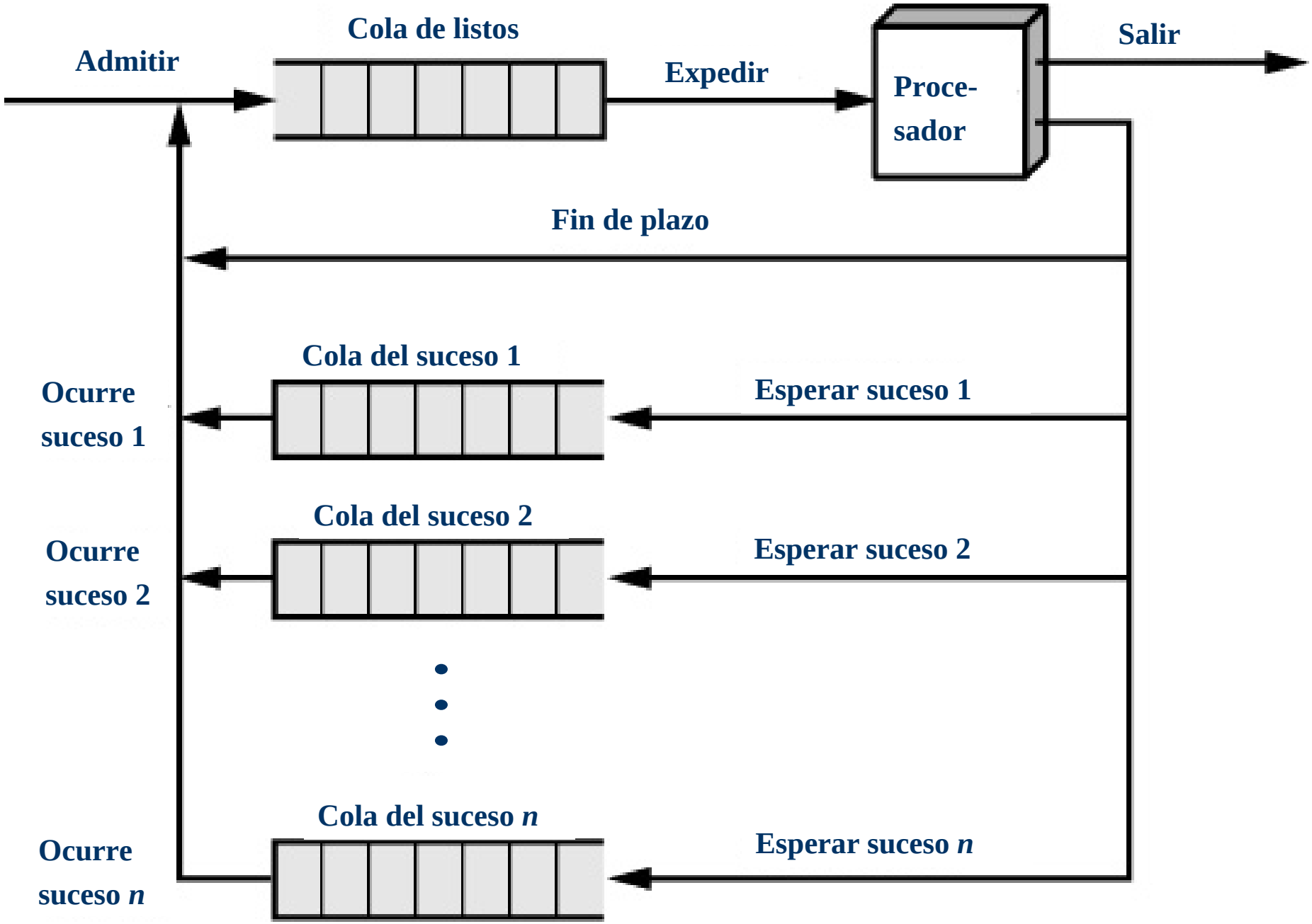


Figura 3.6. Estados de un proceso para la traza de la Figura 3.3.

Normalmente los diagramas no incluyen la actividad el SO (planificador)

Dos colas de bloqueados



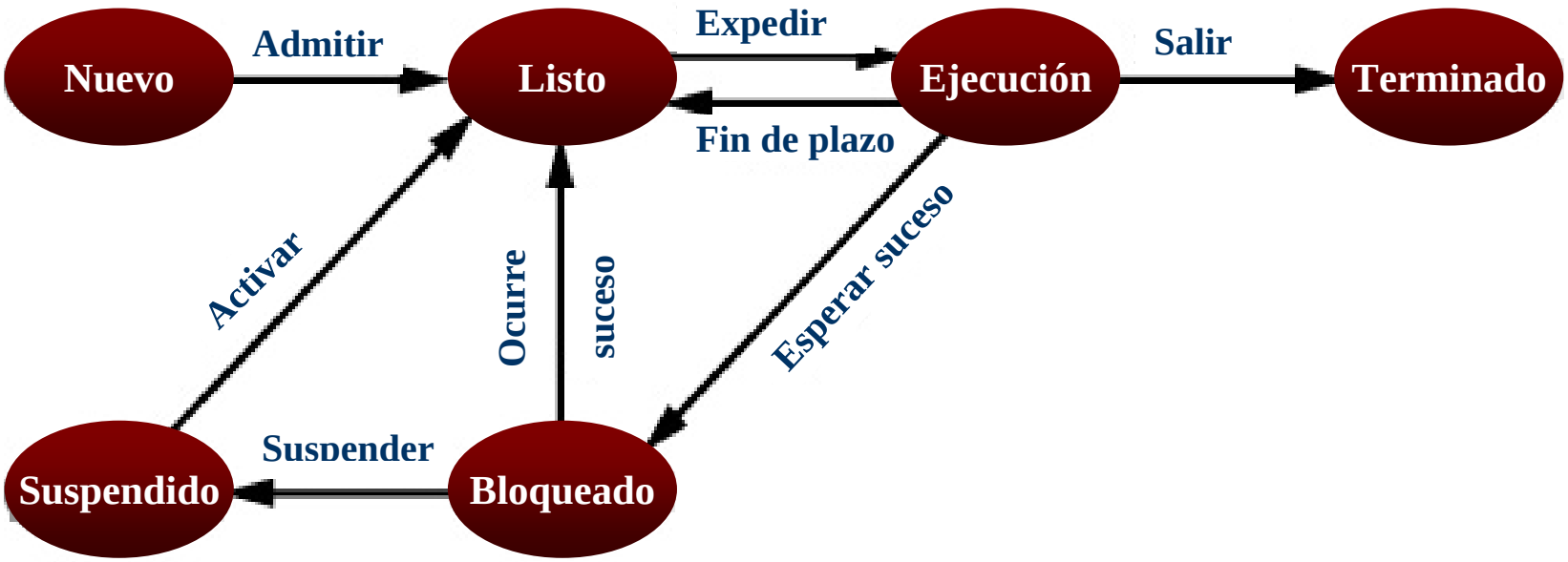


(b) Varias colas de bloqueados

Procesos suspendidos

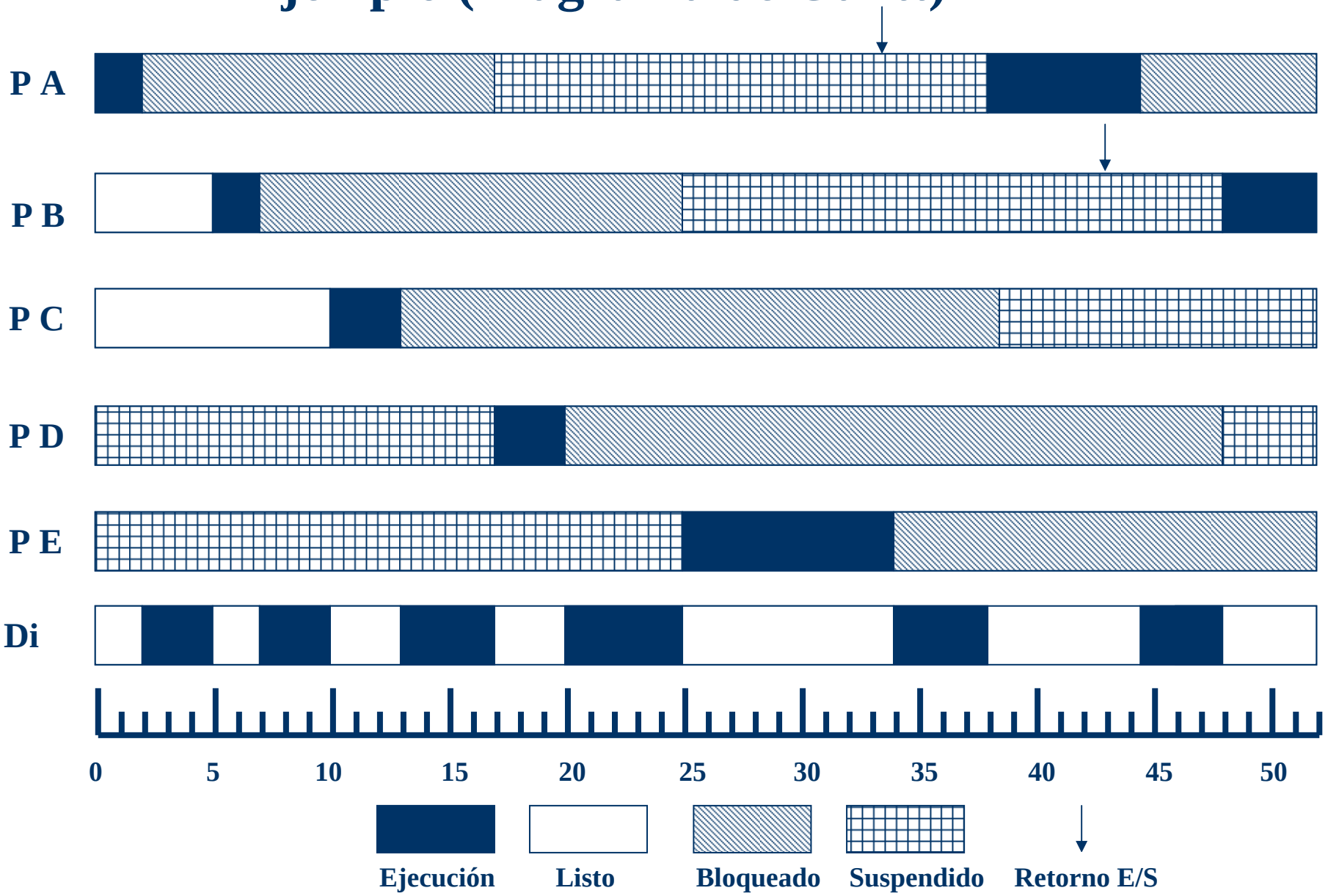
- El procesador es más rápido que la E/S, por lo que suele ser habitual que todos los procesos de memoria estén esperando por E/S.
- Intercambiar una parte (*memoria virtual*) o todo el proceso al disco para liberar memoria ppal.
- Cuando los procesos de la memoria principal están en el estado Bloqueado, el sistema operativo puede suspender un proceso poniéndolo en estado Suspendido.
- Dos nuevos estados:
  - Bloqueado y suspendido.
  - Listo y suspendido.

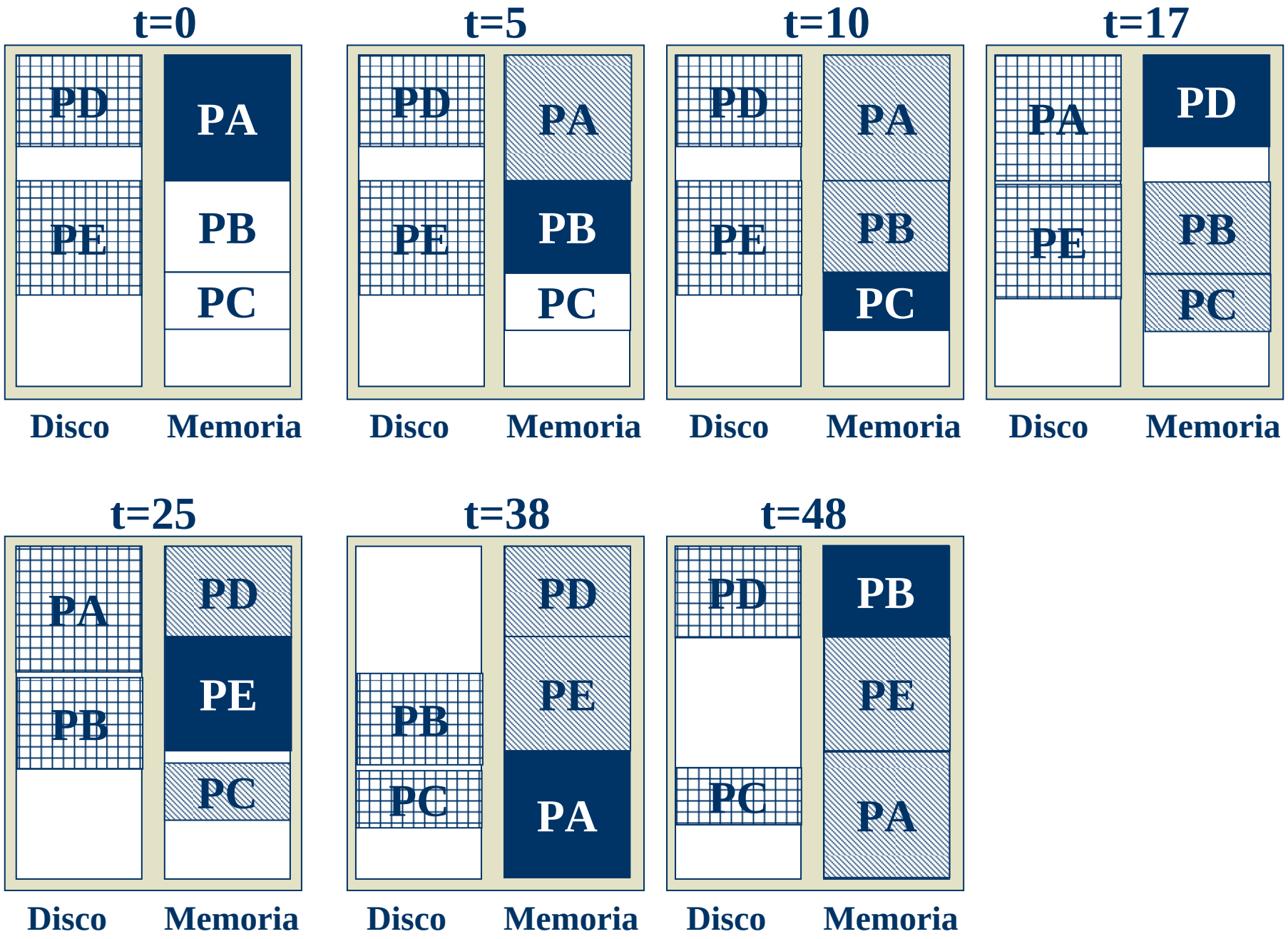
Un estado de suspensión



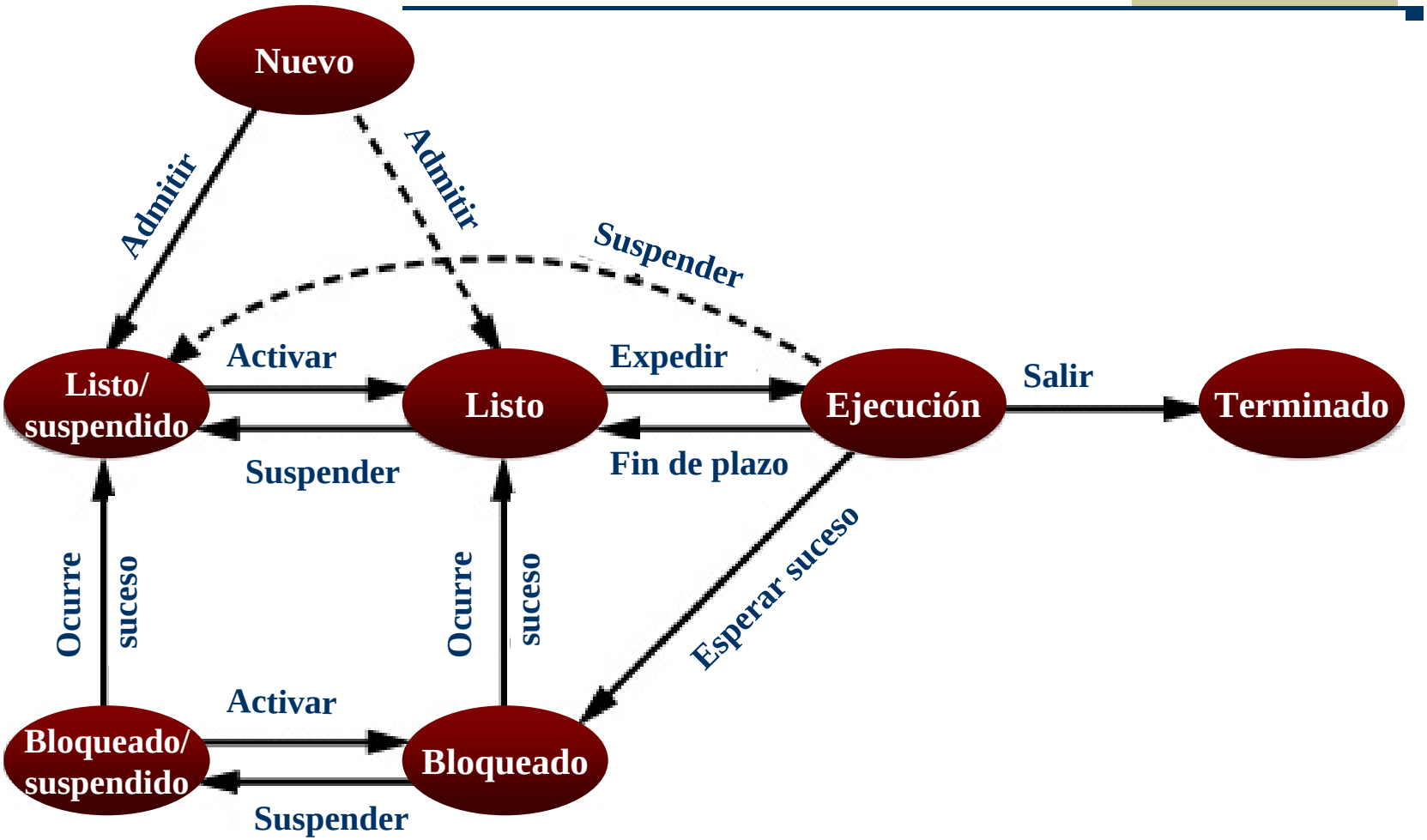
(a) Con un estado de suspensión

Ejemplo (Diagrama de Gantt)




t=0
PD
PE
PA
PB
PC
Disco
Memoria
t=5
PD
PE
PA
PB
PC
Disco
Memoria
t=10
PD
PE
PA
PB
PC
Disco
Memoria
t=17
PA
PE
PD
PB
PC
Disco
Memoria
t=25
PA
PB
PD
PE
PC
Disco
Memoria
t=38
PB
PC
PD
PE
PA
Disco
Memoria
t=48
PD
PC
PB
PE
PA
Disco
Memoria

Dos estados de suspensión



(b) (b) Con dos estados de suspensión

Creación de procesos

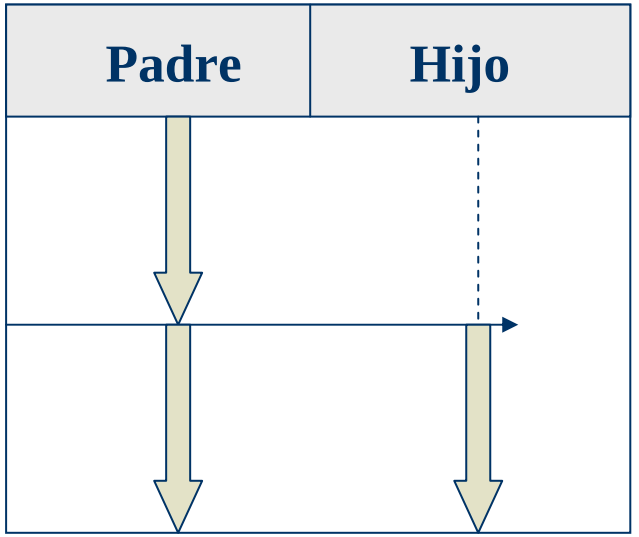
- Emisión de un trabajo por lotes.
- El nuevo usuario intenta conectarse.
- Se crea para ofrecer un servicio, como por ejemplo la impresión.
- Un proceso puede originar la creación de otro.

Creación de procesos

- Proceso hijo inicialmente igual que padre: copia exactamente segmento de datos, pila y estructura de usuario
- Forma de distinguirle del padre: mediante el retorno de fork : 0 para el hijo y el pid del hijo para el padre
- A medida que el hijo va ejecutando otro código, su memoria, pila y estructura se diferencia del padre
- El hijo no termina, hasta que el padre recoge su código de finalización (devuelto con exit).
- El hijo queda zombie, si el padre termina y no ha recogido su código de finalización. El padre debe esperar a sus hijos.
- El hijo queda huérfano, si el padre muere cuando el hijo aún está en ejecución.

Creación de procesos

- Crear nuevo proceso:
  - asignarle entrada en la tabla de procesos y su identificador (pid).
  - estado listo para ejecutar



- Ejemplo:

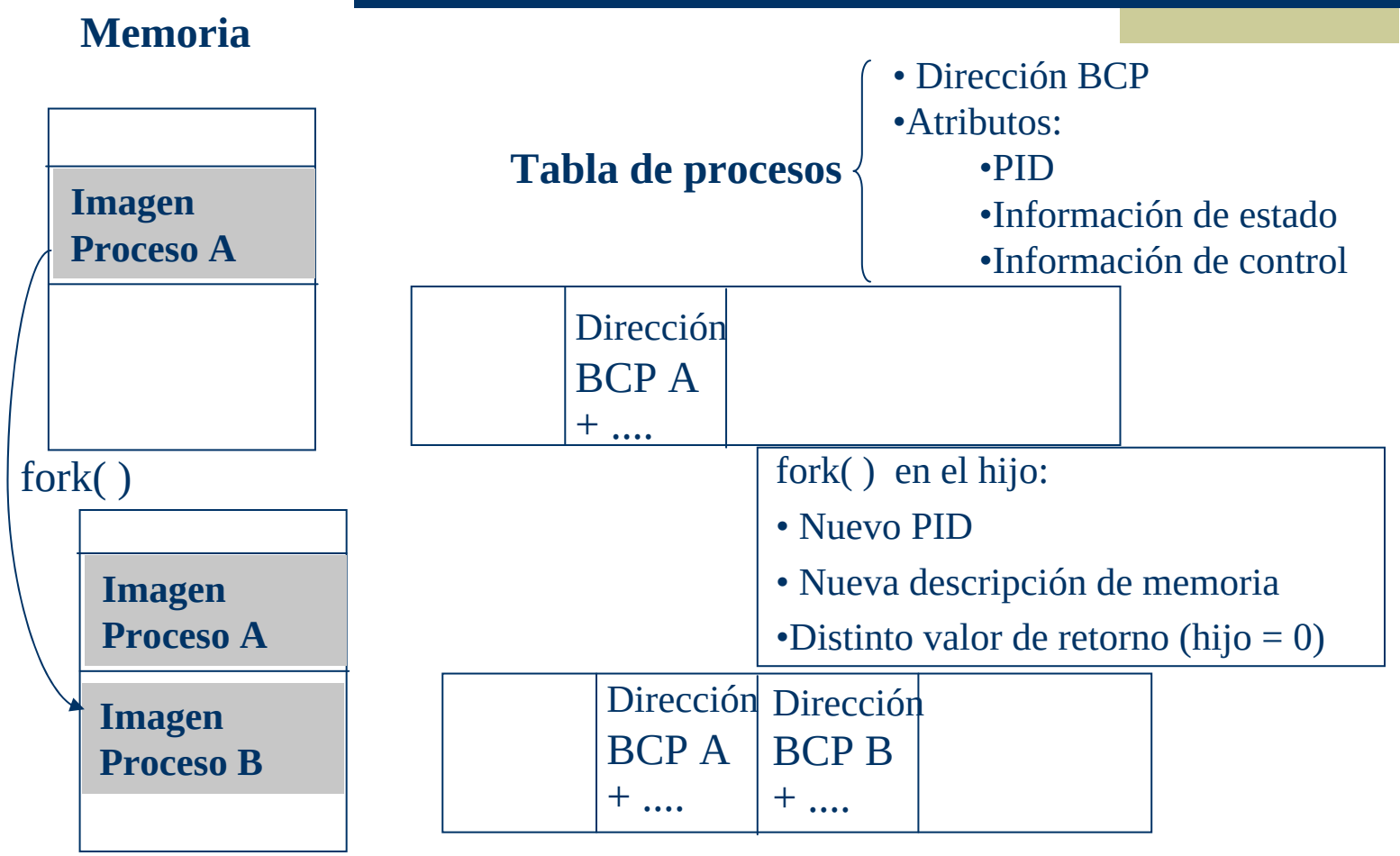
```
int id ;

id = fork();
switch(id)
{
  case -1 : /* error */ break ;
  case 0 : /* hijo */ break ;
  default : /* padre, id tiene el pid del hijo */ break ;
}
```

Creación de procesos

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
   main( )    {
    int childpid;
      ...
     if ((childpid=fork())==0)  fprintf(stderr,"Soy el hijo con PID
                                          %d\n",getpid());
     else if (childpid > 0)   fprintf(stderr,"Soy el padre con PID
                              %d\n",    getpid());
    }
```

- La función fork() devuelve el PID del hijo al padre, devuelve el valor 0 al hijo

Creación de procesos



Creación de procesos

- Asignar una nueva entrada a la tabla principal de procesos.
- Asignar espacio para la imagen del proceso.
- Inicializar el bloque de control de proceso.
- Establecer los enlaces apropiados
  - Ej: Añadir un proceso nuevo a una lista enlazada que se utiliza como cola de planificación.
- Crear o ampliar otras estructuras de datos
  - Ej: Mantener un archivo de contabilidad, actualizar la tabla de gestión de memoria...

Creación de procesos

- Proceso hijo inicialmente igual que el padre salvo en el PID: se copia exactamente el segmento de datos, pila y estructura de usuario.
- El hijo no termina hasta que el padre recoge su código de finalización devuelto con exit( ):
  - El S.O. libera los recursos utilizados, actualiza las estadísticas y lo notifica al padre
- El padre debe esperar a sus hijos, espera con wait( ):
  - detiene al proceso que la invoca hasta que un hijo termine,
  - si no tiene hijos wait( ) regresa de inmediato.

Creación de procesos

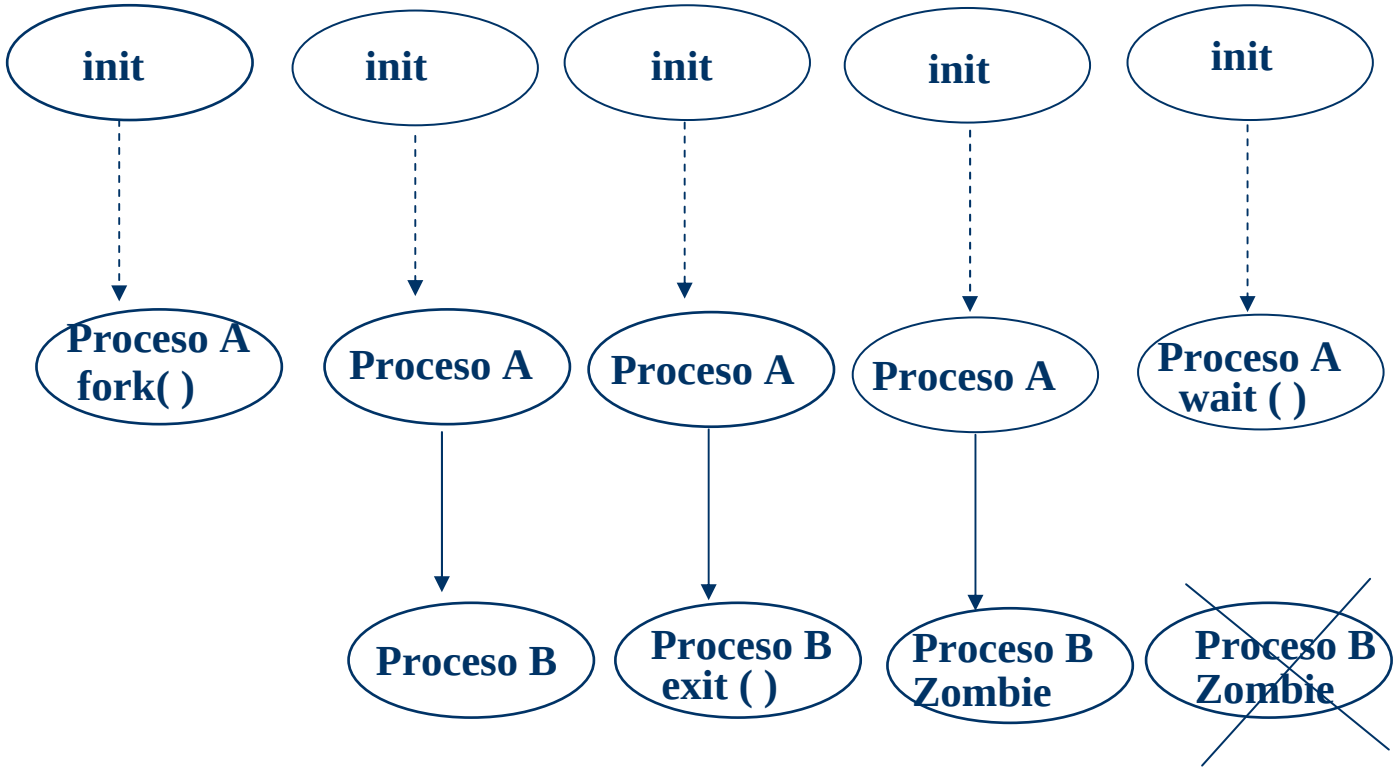
- Forma de distinguir el hijo del padre: mediante el retorno de **fork( )**
  - 0 para el hijo
  - PID del hijo para el padre.
- Tanto el padre como el hijo prosiguen con la ejecución a partir de la siguiente instrucción a la del fork( ). Ambos procesos continúan ejecutando el mismo código.
- Es posible que el proceso hijo reemplace su código inicial por otro código ejecutable y así se diferencie de su padre: A medida que el hijo va ejecutando otro código, su memoria, pila y estructura se diferencian del padre.
  - Familia de llamadas **exec( )** ➔ Permite a un proceso cambiar su imagen de memoria y ejecutar otro programa distinto.

Creación de procesos

- Hijo zombie: Hijo termina y el padre no recoge el código de finalización del hijo.
  - Un proceso zombie no tiene ni área de código, ni área de datos, ni pila, ni estructura ..., pero afecta a la tabla de procesos del sistema, pues sigue ocupando una entrada de la tabla.
  - Los zombies permanecen en el sistema hasta que alguien los “espere”, waitpid( ).

Creación de procesos

Zombie: el hijo muere y el padre no le espera

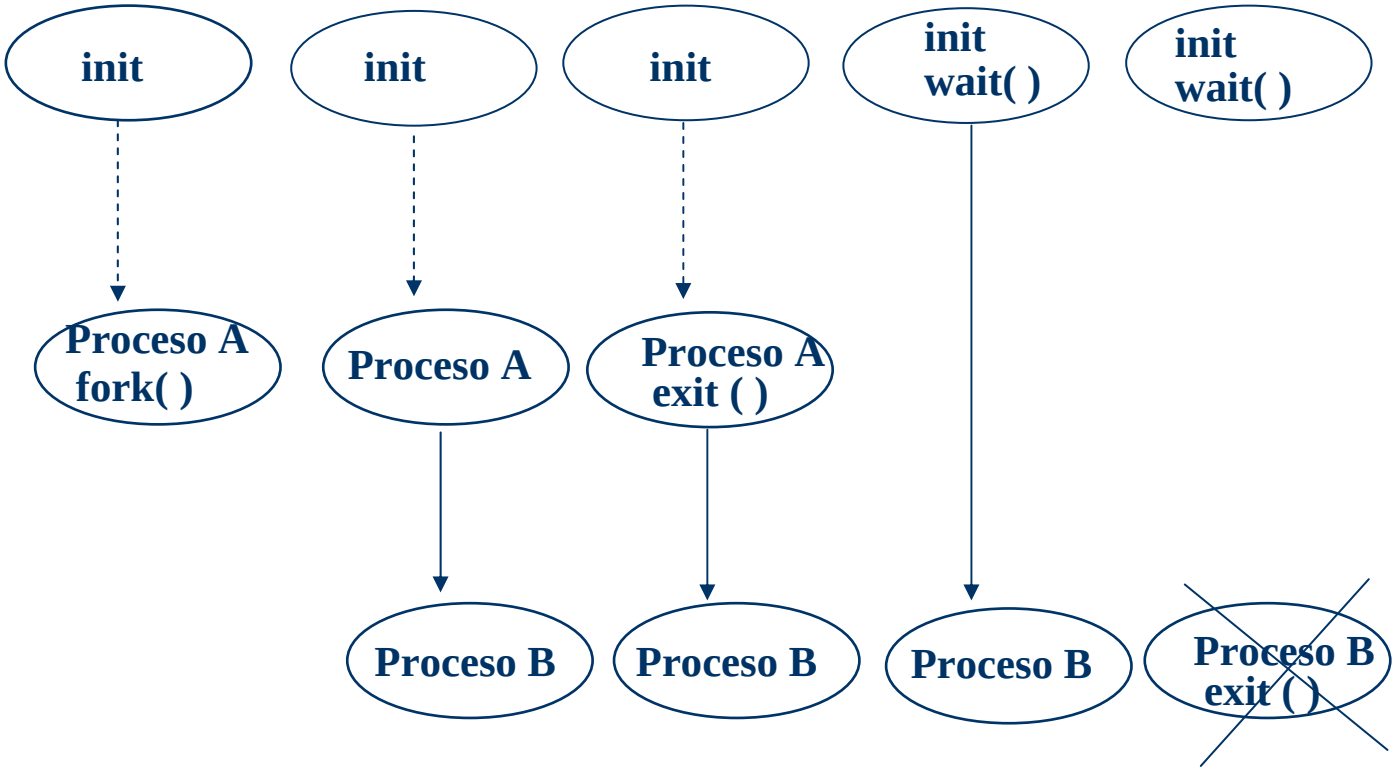


Creación de procesos

- Hijo huérfano: El padre muere antes de que el hijo haya terminado
- Un hijo huérfano es adoptado por el proceso init del sistema.
- El proceso init (PDI=1) es el ancestro de todos los procesos que se crean con posterioridad.
- El proceso con PDI=0, creado por el núcleo cuando arranca el S.O., es el único que no se crea por una llamada a fork( ).

Creación de procesos

El padre muere init adopta a los hijos huérfanos



Terminación de procesos

- Un trabajo por lotes debe incluir una instrucción de detención (*End*).
- El usuario se desconecta.
- El usuario puede abandonar una aplicación.
- Una serie de errores y condiciones de fallo pueden llevarnos a la terminación de un proceso.

Razones para la terminación de un proceso (I)

- Terminación normal.
- Tiempo límite excedido.
- No hay memoria disponible.
- Violación de límites.
- Error de protección:
  - Por ejemplo: escribir en un archivo que es sólo de lectura.
- Error aritmético.
- Tiempo máximo de espera rebasado:
  - El proceso ha esperado más allá del tiempo máximo especificado para que se produzca cierto suceso.

Razones para la terminación de un proceso (II)

- Fallo de E/S.
- Instrucción ilegal
- Instrucción privilegiada.
- Mal uso de los datos.
- Intervención del operador o del SO:
  - Por ejemplo, si se produce un bloqueo.
- Terminación del padre, por lo que terminan los procesos de todos sus descendientes.
- Solicitud del padre.

Razones para la suspensión de procesos

Intercambio	El sistema operativo necesita liberar suficiente memoria principal para cargar un proceso que está listo para ejecutarse.
Otra razón del SO	El sistema operativo puede suspender a un proceso subordinado o de utilidad, o a un proceso que se sospecha que sea el causante de un problema.
Solicitud de un usuario interactivo	Un usuario puede querer suspender la ejecución de un programa con fines de depuración o en relación con el uso de un recurso.
Temporización	Un proceso puede ejecutarse periódicamente (por ejemplo, un proceso de contabilidad o de supevisión del sistema) y puede ser suspendido mientras espera el siguiente intervalo de tiempo.
Solicitud del proceso padre	Un proceso padre puede querer suspender la ejecución de un descendiente para examinar o modificar el proceso suspendido o para coordinar la actividad de varios descendientes.

Descripción de procesos

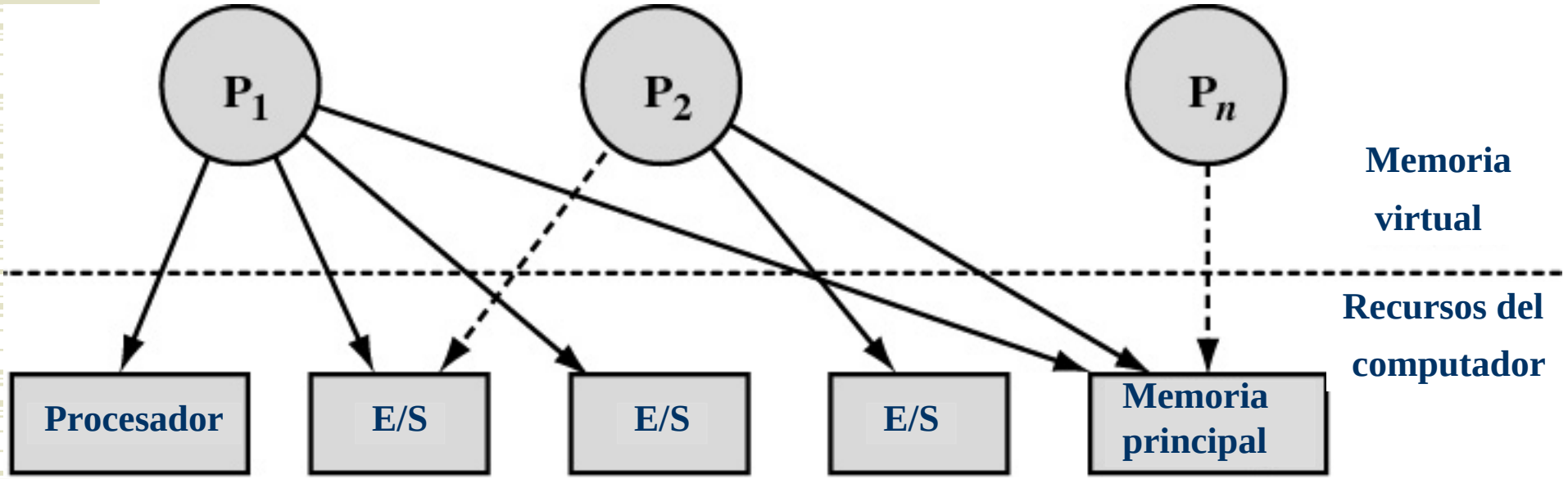


Figura 3.9. Procesos y recursos (asignación de recursos en un instante de tiempo).

Estructuras de control del sistema operativo

- Información sobre el estado actual de cada proceso y de cada recurso.
- El sistema operativo construye tablas de información sobre cada entidad que esté administrando.

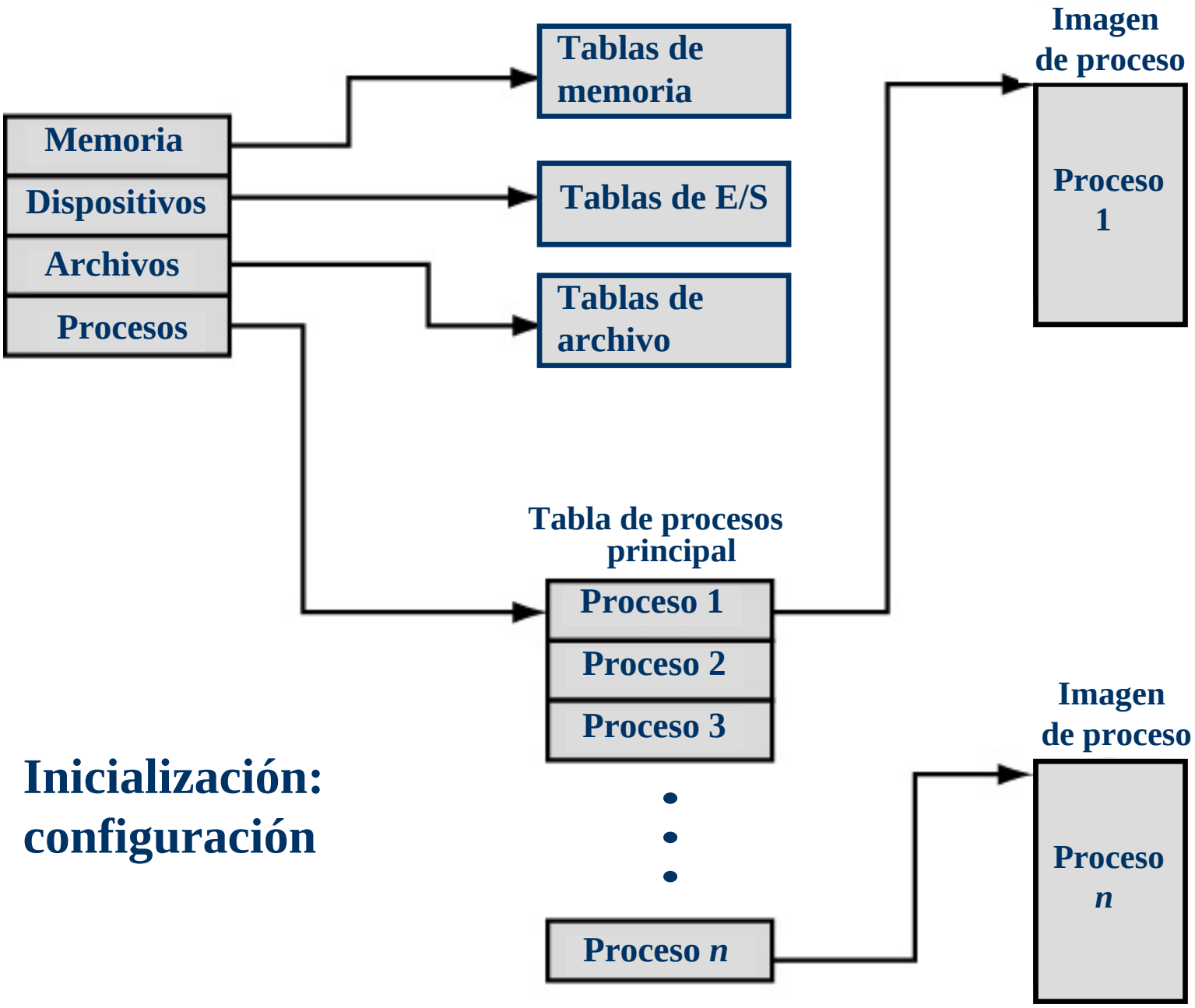


Figura 3.10. Estructura general de las tablas de control del sistema operativo.

Tablas de memoria

- La asignación de memoria principal a los procesos.
- La asignación de memoria secundaria a los procesos.
- Atributos de protección de bloques de memoria principal o virtual, como qué procesos pueden acceder a ciertas regiones compartidas de memoria.
- Cualquier información necesaria para gestionar la memoria virtual.

Tablas de E/S

- Un dispositivo de E/S puede estar disponible o estar asignado a un proceso en particular.
- Estado de la operación de E/S.
- Posición de memoria principal que se está utilizando como origen o destino de la transferencia de E/S.

Tablas de archivos

- Ofrecen información sobre la existencia de los archivos.
- Su posición en la memoria secundaria.
- Su estado actual.
- Otros atributos.
- A veces esta información es mantenida por un sistema de gestión de archivos.

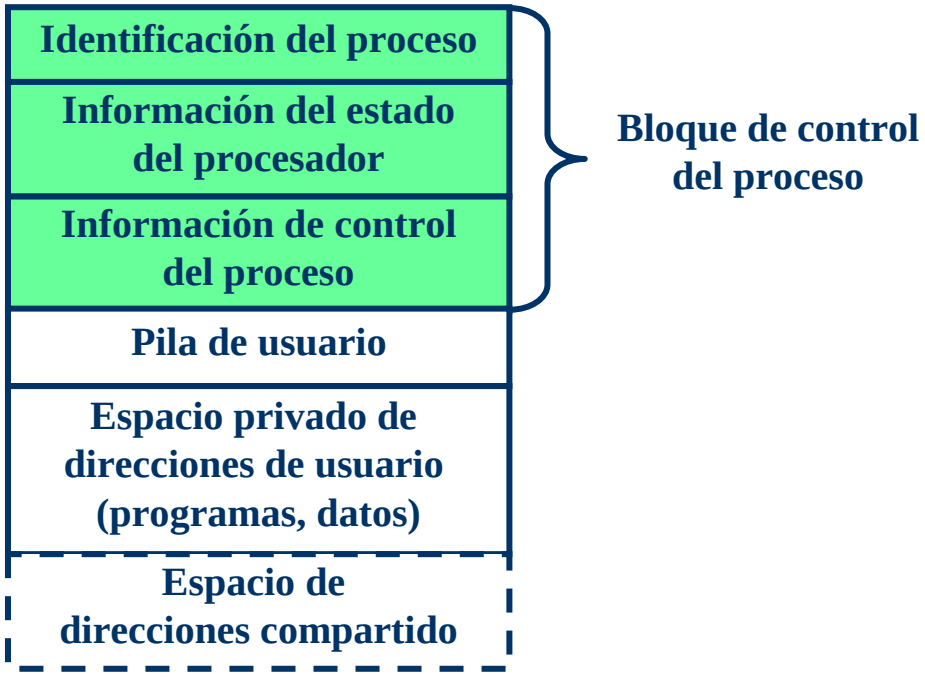
Tablas de procesos

- ¿Dónde está ubicado el proceso?
- Atributos del proceso necesarios para su administración:
  - ID del proceso.
  - Estado del proceso.
  - Ubicación en la memoria.

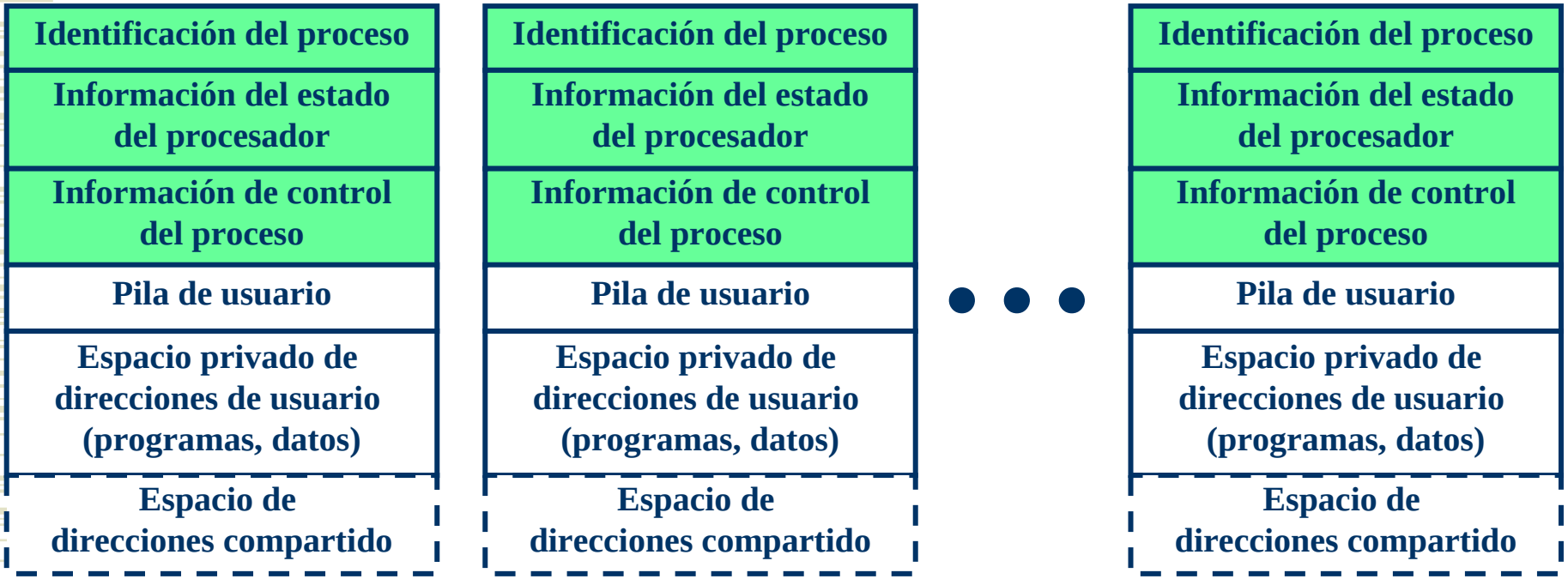
Ubicación de los procesos

- Un proceso incluye un programa o un conjunto de programas:
  - **Instrucciones** a ejecutar
  - **Datos** de usuario, variables locales y globales, constantes definidas.
  - **Pila** para almacenamiento de parámetros y direcciones de retorno de procedimientos
  - **Bloque de control del proceso**:
    colección de atributos para el SO, para que controle el proceso

Imagen de un proceso



Procesos en memoria virtual



Bloque de control de proceso

- Identificación de proceso
  - Identificadores numéricos:
    - Identificador de este proceso.
    - Identificador del proceso que creó a este proceso (el proceso padre).
    - Identificador del usuario.

UID	PID	PPID							
root	247	1	0	Feb	11	?		0:00	/usr/lib/lpsched
root	167	1	0	Feb	11	?		0:01	/usr/sbin/rpcbind
root	395	392	0	Feb	11	?		0:00	/usr/lib/saf/ttymon
root	196	1	0	Feb	11	?		0:00	/usr/sbin/inetd -s
root	201	1	0	Feb	11	?		0:00	/usr/lib/nfs/lockd
daemon	202	1	0	Feb	11	?		0:00	/usr/lib/nfs/statd
root	280	1	0	Feb	11	?		0:00	/usr/sbin/ifbdaemon /dev/fbs/ifb0
root	260	1	0	Feb	11	?		0:10	/usr/lib/print/printd
root	269	1	0	Feb	11	?		0:00	/usr/sadm/lib/smc/bin/smcboot
root	270	269	0	Feb	11	?		0:00	/usr/sadm/lib/smc/bin/smcboot
root	282	1	0	Feb	11	?		0:02	/usr/sbin/vold
root	357	1	0	Feb	11	?		0:01	/usr/local/bin/ntpd -A -c /etc/ntp.conf -l /var/log/ntp.log
root	351	1	0	Feb	11	?		0:00	/usr/lib/dmi/snmpXdmid -s meneceo
daemon	300	1	0	Feb	11	?		0:00	/usr/lib/ab2/dweb/sunos5/bin/dwhttpd /usr/lib/ab2/dweb/data
daemon	301	300	0	Feb	11	?		0:00	/usr/lib/ab2/dweb/sunos5/bin/dwhttpd /usr/lib/ab2/dweb/data
carlos	396	344	3	Feb	11	?	312:02	00	6 /usr/openwin/bin/Xsun :0 -nobanner -auth /var/dt/A:0-lNaaRa
root	340	1	0	Feb	11	?		0:00	/usr/lib/snmp/snmpdx -y -c /etc/snmp/conf
root	393	1	0	Feb	11	conso	0:00		/usr/lib/saf/ttymon -g -h -p meneceo console login: -T sun -
root	344	1	0	Feb	11	?		0:00	/usr/dt/bin/dtlogin -daemon
root	389	1	0	Feb	11	?		0:02	/usr/ssh/sbin/sshd
root	397	340	0	Feb	11	?		1:04	mibiisa -r -p 32797
root	350	1	0	Feb	11	?		0:00	/usr/lib/dmi/dmispd

UID: Identificador de Usuario

PID: Identificador del Proceso

PPID: Identificador del Proceso Padre

Bloque de control de proceso

- Información de estado del procesador
  - Registros visibles para el usuario:

    Un registro visible para el usuario es aquél al que puede hacerse referencia por medio del lenguaje de máquina que ejecuta el procesador. Normalmente, existen de 8 a 32 de estos registros, aunque algunas implementaciones RISC tienen más de 100.

Ejemplo, en lenguaje C:

```
register int i;
for(i=0;i<10000;++i) {
}
```

Bloque de control de proceso

- Información de estado del procesador (cont)
  - Registros de control y de estado:

    Hay varios registros del procesador que se emplean para controlar su funcionamiento. Entre éstos se incluyen:

    - *Contador de programa*: contiene la dirección de la próxima instrucción a leer.
    - *Códigos de condición*: muestran el resultado de la operación aritmética o lógica más reciente (signo, cero, acarreo, igualdad, desbordamiento).
    - *Información de estado*: incluye los indicadores de habilitación o inhabilitación de interrupciones y de modo de ejecución.

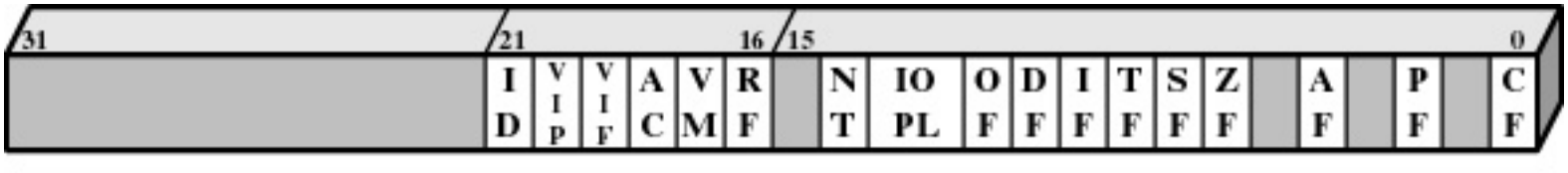
Bloque de control de proceso

- Información de estado del procesador (cont)
  - Punteros de pila:
    - Cada proceso tiene una o más pilas LIFO del sistema asociadas. Las pilas se utilizan para almacenar los parámetros y las direcciones de retorno de los procedimientos y de las llamadas al sistema. El puntero de pila siempre apunta a la cima de la pila.

Información de estado del procesador

- Formada por el contenido de los registros del procesador:
  - Registros visibles para el usuario.
  - Registros de control y de estado.
  - Punteros de pila.
- Palabra de estado del programa (PSW):
  - Contiene información de estado.
  - Por ejemplo: el registro EFLAGS de las máquinas Pentium.

Registro EFLAGS del Pentium II



ID	=	Marca de identificación	DF	=	Marca de dirección
VIP	=	Interrupción virtual pendiente	IF	=	Marca de inhabilitación de interrupciones
VIF	=	Marca de interrupción virtual	TF	=	Marca de cepo
AC	=	Comprobación de alineación	SF	=	Marca de signo
VM	=	Modo 8086 virtual	ZF	=	Marca de cero
RF	=	Marca de continuación	AF	=	Marca de acarreo auxiliar
NT	=	Marca de tarea anidada	PF	=	Marca de paridad
IOPL	=	Nivel de privilegio de E/S	CF	=	Marca de acarreo
OF	=	Marca de desbordamiento			

Figura 3.11. Registro EFLAGS del Pentium II.

Bloque de control de proceso

- Información de control del proceso
  - Información de planificación y de estado:

    La necesita el sistema operativo para llevar a cabo planificación. Elementos típicos:

    •*Estado del proceso*: define la disposición del proceso para ser planificado para ejecutar (en ejecución, listo, esperando, detenido).

    •*Prioridad*: se puede usar uno o más campos para describir la prioridad de planificación de los procesos. En algunos sistemas se necesitan varios valores (por omisión, actual, la más alta permitida).

    •*Información de planificación*: ésta dependerá del algoritmo de planificación utilizado. Ejs: cantidad de tiempo que el proceso ha estado esperando y cantidad de tiempo que el proceso ejecutó la última vez.

    •*Suceso*: la identidad del suceso que el proceso está esperando antes de poder reanudarse.

Bloque de control de proceso

- Información de control del proceso (cont):
  - Estructuración de datos:
    - Un proceso puede estar enlazado con otros procesos en una cola, un anillo o alguna otra estructura.
    - Ej: Procesos en espera de un nivel determinado de prioridad, enlazados en una cola.
    - Un proceso puede mostrar una relación padre-hijo (creador-creado) con otro proceso.
    - El bloque de control de proceso puede contener punteros a otros procesos para dar soporte a estas estructuras.

Bloque de control de proceso

- Información de control del proceso (cont):
  - Comunicación entre procesos:
    - Puede haber varios indicadores, señales y mensajes asociados con la comunicación entre dos procesos independientes. Una parte de esta información o toda ella se puede guardar en el bloque de control de proceso.
  - Privilegios de los procesos:
    - A los procesos se les otorgan privilegios en términos de la memoria a la que pueden acceder y el tipo de instrucciones que pueden ejecutar. Además, también se pueden aplicar privilegios al uso de los servicios y utilidades del sistema.

Bloque de control de proceso

- Información de control del proceso:
  - Gestión de memoria:
    - Esta sección puede incluir punteros a las tablas de páginas o segmentos que describen la memoria virtual asignada al proceso.
  - Propiedad de los recursos y utilización:
    - Se pueden indicar los recursos controlados por el proceso, como los archivos abiertos. También puede incluir un historial de la utilización del procesador o de otros recursos ; esta información puede ser necesaria para el planificador.

Control de Procesos
Modos de ejecución

- Modo de usuario:
  - Es el modo menos privilegiado.
  - Los programas de usuarios ejecutan normalmente en ese modo.
- Modo del sistema, modo de control o modo del núcleo:
  - Es el modo más privilegiado.
  - Núcleo del sistema operativo.

Cuándo cambiar de proceso

- Interrupción de reloj:
  - El proceso en ejecución ha consumido la fracción máxima de tiempo permitida.
- Interrupción de E/S:
  - Procesos bloqueados pasan a listos (suspendidos => suspendidos y listos)
- Fallo de memoria:
  - La dirección de memoria se encuentra en la memoria virtual, por lo tanto debe ser llevada a la memoria principal (proceso pasa a bloqueado, esperando E/S).

Cuándo cambiar de proceso

- Cepos/Trap:
  - Se ha producido un error.
  - Puede hacer que el proceso que se estaba ejecutando pase al estado de Terminado (si fatal). Depende del SO
- Llamada del supervisor:
  - Ej: abrir un archivo. Transferencia a rutina que forma parte del SO => proceso de usuario bloqueado

Cambio de modo

- **Salvar el contexto** del programa que se ejecuta
- Asignar al **contador** de programa valor de la dirección de comienzo de programa tratamiento de interrupción
- Cambiar de modo usuario a modo **núcleo**, para que en el procesamiento de insterrupción pueda haber **instrucciones privilegiadas**.
- **Ejecutar rutina** de tratamiento de interrupción.

Nota: es siempre que llega una interrupción, no sólo para el cambio de proceso

Cambio de modo

- Cambio de modo no implica posterior cambio de estado del proceso que se estaba ejecutando
- Se puede cambiar de modo, ejecutar instrucciones privilegiadas en modo núcleo y volver a ejecutar el mismo proceso
  => no es necesario hacer cambio completo de proceso
  => basta con guardar info del estado del procesador

Cambio de estado de los procesos

- **Salvar el contexto** del procesador, incluyendo el contador de programa y otros registros.
- **Actualizar el bloque** de control del proceso que está en estado de Ejecución.
- **Mover** el bloque de control del proceso a la cola apropiada (Listos, bloqueados).
- **Seleccionar** otro proceso para su ejecución.

Cambio de estado de los procesos

- Actualizar el **bloque** de control del proceso seleccionado.
- Actualizar las **estructuras** de datos de la gestión de memoria.
- Restaurar el **contexto** del proceso seleccionado.

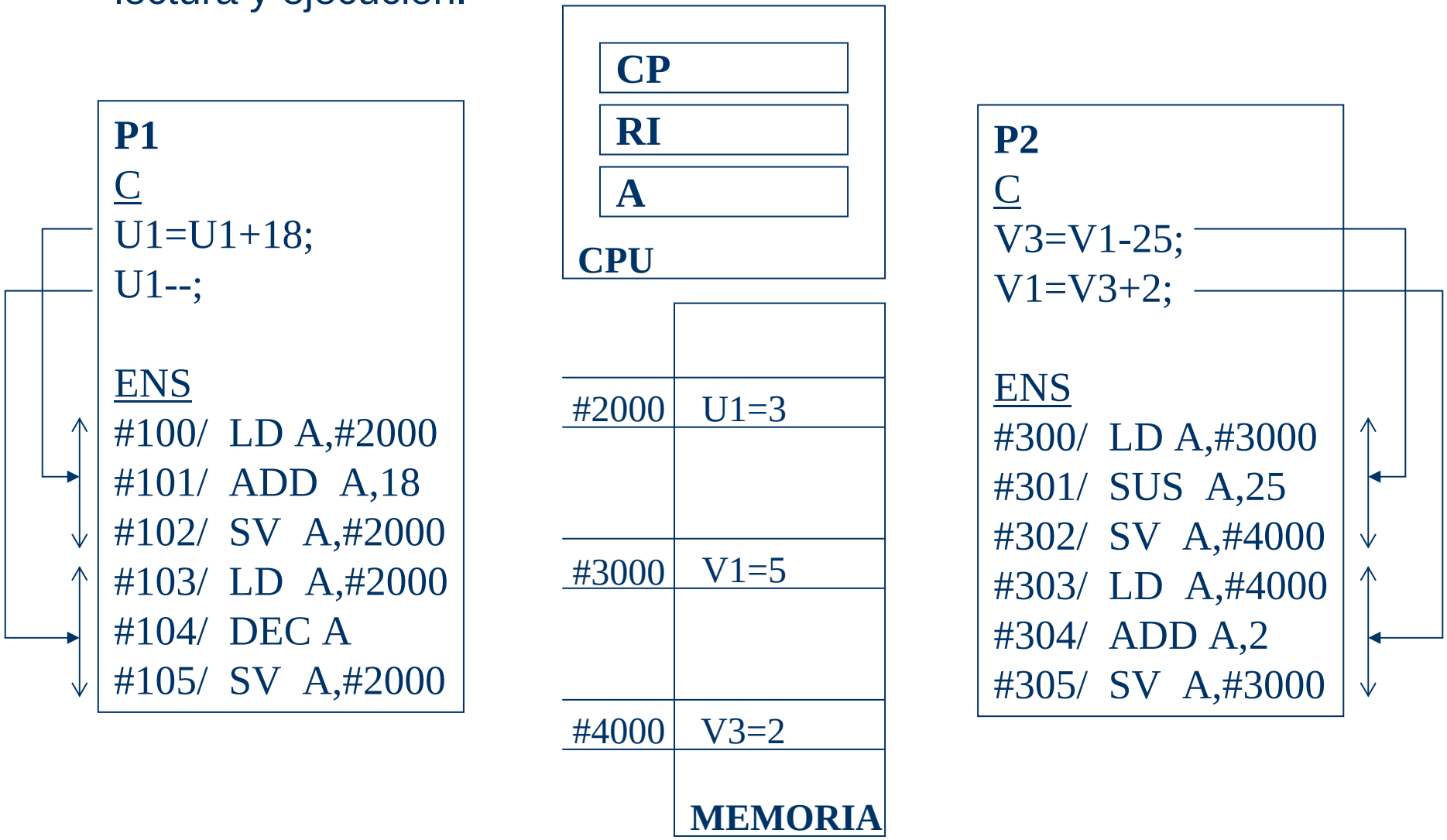
Cambio de estado de los procesos (y III)

- ¿Quién realiza las operaciones anteriores para realizar un cambio de estado del proceso?
  - El sistema operativo
- ¿Qué cambios hacen falta entonces?
  1. Cambio de modo: a modo núcleo
  2. S.O. realiza cambio de proceso
  3. Cambio de modo: a modo usuario

Cambio de proceso

- **Salvar contexto** de P1 (BCP)
- Asignar a CP la dirección de **comienzo de** programa **tratamiento de interrupción de cambio de proceso**
- Cambiar a **modo núcleo**
- **Ejecutar rutina** tratamiento de interrupción (cambio de proceso)
  - **Actualizar el bloque de control del proceso** que está en estado de ejecución (nuevo estado: listo/bloqueado/…)
  - **Mover** el bloque de control del proceso **a la cola apropiada** (listos/bloqueados/)
  - **Seleccionar otro proceso** para su ejecución.
  - Actualizar BCP seleccionado (nuevo estado: ejecución)
  - Actualizar estructuras de datos de la gestión de memoria (traducción de direcciones)
  - Restaurar contexto del proceso seleccionado (actualizar registros del procesador)
- Cambiar a **modo usuario**

- La unidad de trabajo (ininterrumpible) de la CPU es la instrucción en código máquina (o ensamblador), no la instrucción de lenguaje en alto nivel. Hay un ciclo de interrupción después de cada ciclo de lectura y ejecución.



	RI	CP	A	U1	V1	V3
	SO	#100	?	3	5	2
	#100	#101	3	3	5	2
Int	#101	#102	21	3	5	2
	SO	#300	?	3	5	2
	#300	#301	5	3	5	2
Int						
Int						
Int						

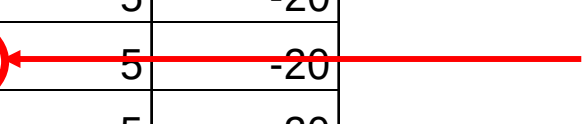
Guarda A=21 en BCP1

	RI	CP	A	U1	V1	V3	
	SO	#100	?	3	5	2	
	#100	#101	3	3	5	2	
Int	#101	#102	21	3	5	2	→ Guarda A=21 en BCP1
	SO	#300	?	3	5	2	
	#300	#301	5	3	5	2	
	#301	#302	-20	3	5	2	→ Guarda A=-20 en BCP2
Int	#302	#303	-20	3	5	-20	← Carga A=21 de BCP1
	SO	#102	21	3	5	-20	
	#102	#103	21	21	5	-20	
	#103	#104	21	21	5	-20	
Int	#104	#105	20	21	5	-20	→ Guarda A=20 en BCP1
	SO	#303	-20	21	5	-20	← Carga A=-20 de BCP2
	#303	#304	-20	21	5	-20	
Int	#304	#305	-18	21	5	-20	→ Guarda A=-18 en BCP2
	SO	#105	20	21	5	-20	← Carga A=20 de BCP1
	#105	Int	20	20	5	-20	
	SO	#305	-18	20	5	-20	
	#305	**Int**	**-18**	**20**	**-18**	**-20**	

Ejercicio: ¿Qué ocurre si no hay cambio de contexto?

	RI	CP	A	U1	V1	V3
	SO	#100	?	3	5	2
	#100	#101	3	3	5	2
Int	#101	#102	21	3	5	2
	SO	#300	21	3	5	2
	#300	#301	5	3	5	2
Int						
Int						
Int						

	RI	CP	A	U1	V1	V3
	SO	#100	?	3	5	2
	#100	#101	3	3	5	2
Int	#101	#102	21	3	5	2
	SO	#300	21	3	5	2
	#300	#301	5	3	5	2
	#301	#302	-20	3	5	2
Int	#302	#303	-20	3	5	-20
	SO	#102	-20	3	5	-20
	#102	#103	-20	-20	5	-20
	#103	#104	-20	-20	5	-20
Int	#104	#105	-21	-20	5	-20
	SO	#303	-21	-20	5	-20
	#303	#304	-20	-20	5	-20
Int	#304	#305	-18	-20	5	-20
	SO	#105	-18	-20	5	-20
	#105	Int	-18	-18	5	-20
	SO	#305	-18	-18	5	-20
	#305	**Int**	**-18**	**-18**	**-18**	**-20**



Ejecución del sistema operativo

- S.O. es programa que se ejecuta
- ¿Relación con procesos de usuario en cuanto a ejecución (dónde y en qué modo se ejecutan)?

    1. Núcleo fuera de todo proceso
    2. Ejecución dentro de los procesos de usuario
    3. Sistema operativo basado en procesos

Ejecución del sistema operativo

- Núcleo fuera de todo proceso:
  - Ejecuta el núcleo del sistema operativo fuera de cualquier proceso.
  - El código del sistema operativo se ejecuta como una entidad separada que opera en modo privilegiado.
- Ejecución dentro de los procesos de usuario:
  - Software del sistema operativo en el contexto de un proceso de usuario.
  - Un proceso se ejecuta en modo privilegiado cuando se ejecuta el código del sistema operativo.

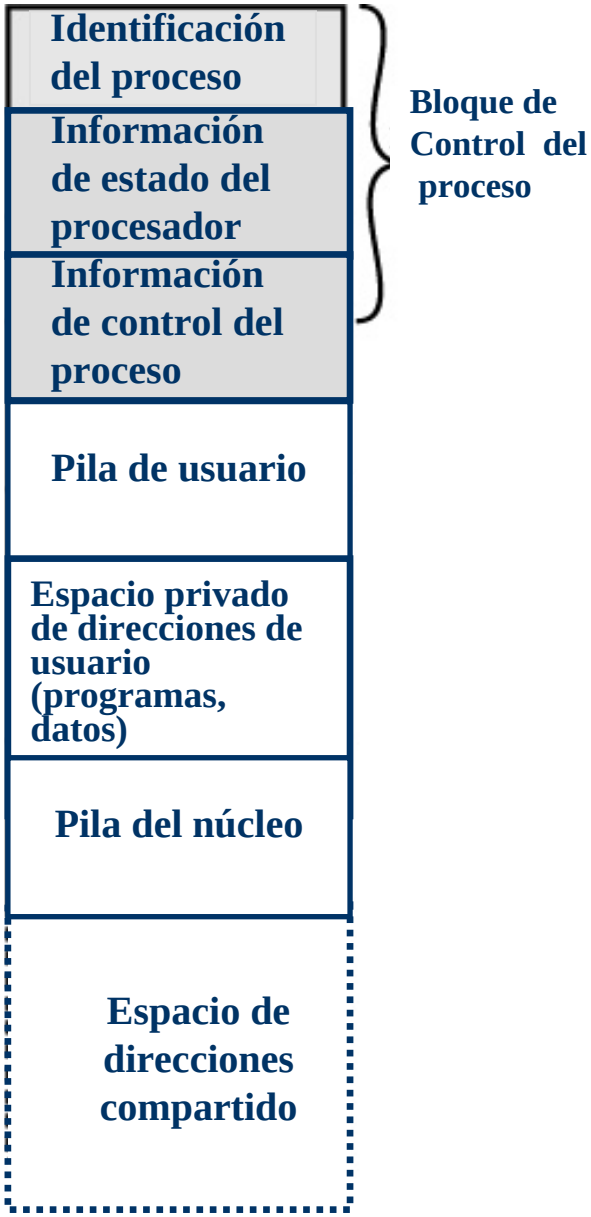


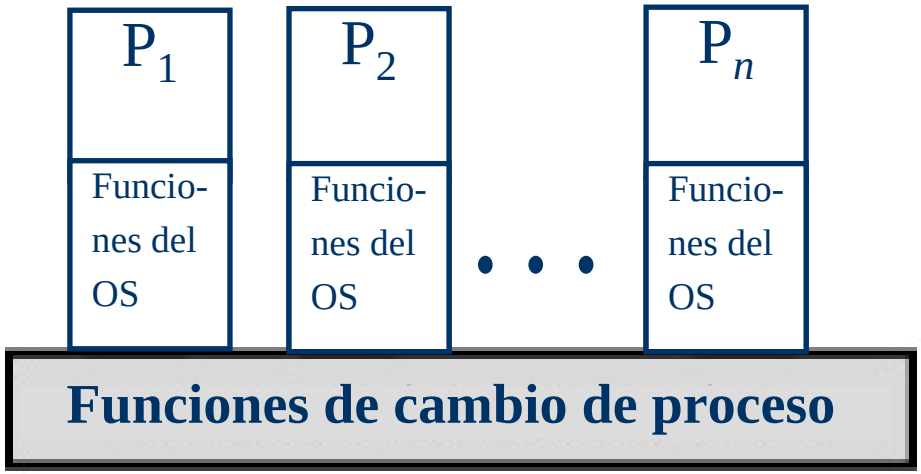
Figura 3.15. Imagen de un proceso: el sistema operativo se ejecuta dentro del proceso de usuario.

Ejecución del sistema operativo

- Sistema operativo basado en procesos:
  - Las funciones más importantes del núcleo se organizan en procesos separados.
  - Útil en un entorno de multiprocesador o de varios computadores.

Gestión de procesos en UNIX SVR4

- La mayoría del sistema operativo ejecuta dentro de un proceso de usuario.



(b) Las funciones del SO se ejecutan dentro de los procesos de usuario

Próximo tema

- Stallings 4.1 y 4.2